

colorlinks=true, linkcolor=blue, filecolor=magenta, urlcolor=cyan, cite-
color=blue,



The role of machine learning in personalised instructional sequencing for language learning

Ahmed Hasan Zaidi



King's College

This dissertation is submitted on November, 2020 for the degree of Doctor of Philosophy

Abstract

The origins of personalised instructional sequencing can be dated back to the times of the Ancient Greeks to the times of Alexander The Great's tutor, Aristotle. However, over the centuries the demand for education and growth of students has been disproportionately greater than the number of teachers in training. Therefore, there has been a longstanding interest in finding a way to scale education without negatively affecting learning outcomes. This interest was fuelled further with the advent of computers and artificial intelligence, where a plethora of systems and models were built to bring technology driven personalised instructional sequencing to the world. Unfortunately, results were far from groundbreaking and many challenges still remain.

In my thesis, I investigate three aspects of personalised instructional sequencing: the personalised instructional sequencing mechanism, the student knowledge representation, and human forgetting. While I do not cover the entirety of personalised instructional sequencing, I cover what I consider the foundational components. I link psychological theory to model selection and design in each of my systems and present experiments to illustrate their impact. I show how reinforcement learning can be used for vocabulary learning. I also present a model that uses neural collaborative filtering to learn student knowledge representations. Lastly, I present a state-of-the-art model to predict the probability of vocabulary word recall for students learning English as a second language. The system's novelty lies in the use of word complexity to adapt the forgetting curve as well as its incorporation of psychological theory to select an appropriate model.

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any that I have submitted, or am concurrently submitting, for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or is being concurrently submitted, for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. This dissertation does not exceed the prescribed limit of 60 000 words.

Ahmed Hasan Zaidi

November, 2020

Acknowledgements

Firstly, I would like to thank God for giving me the strength to get through this chapter of my life. Although it has been a long journey with many hurdles, it has been a journey that has given me the opportunity to get to know myself and others in a way that would otherwise not be possible.

I would like to thank my supervisor Paula Buttery for her endless patience and support during my PhD. I would also like to thank Andrew Caines, Russell Moore, Mark Elliot and Andrew Rice for being incredible collaborators and travel partners. I am also grateful to ALTA, King's College and the Computer Laboratory for always making me feel welcome. I am very grateful to my examiners Michael Mozer and Simone Teufel who went through the trouble to read this thesis and provide me with valuable feedback.

During my PhD, I have also been fortunate to meet and receive the support of people outside my department to whom I am very grateful to. In no particular order, I would like to thank my team at Catalyst AI, Raymond Siems, Sajjad Haider and Evan Martin who always backed me during the intense periods of my PhD. I would like to thank the Woolf Institute for giving me another home and allowing me to satisfy my intellectual interests beyond computer science.

I would also like to mention my friends, who although never understood what I did during my PhD, always provided me with boundless support and motivation including Zohaib Raza, Talal Khan, Timmy Khan, Naufil Daar, Moe Bunny, Omar Ilyas and of course Linnéa Strand. Each one of them in their unique way acted like my guardians, ensuring I maintained laser focus throughout my PhD.

Finally, I would like to thank my family. My parents, Tanwir and Uzma who never

wavered in their support and belief in me. My siblings, Salman, Ammar, Sarah and Mohammed who reminded me of the important things in life. And my niece Nora, who came into this world during the final year of my PhD and has been a shining light for all of us. I hope one day she is able to read and understand and build upon the work described in this thesis.

Contents

1	Introduction	15
1.1	Topical navigation	24
1.2	Experimental chapter summary	25
2	Background	27
2.1	Instructional Sequencing	27
2.1.1	Origins of instructional sequencing	27
2.1.2	Emergence of scientific instructional sequencing	30
2.1.3	Computational instructional sequencing	33
2.1.4	Artificial intelligence and instructional sequencing	37
2.2	Knowledge representation	39
2.3	Memory and forgetting	44
3	Instructional sequencing for vocabulary learning	47
3.1	Introduction	47
3.1.1	The notion of difficulty	49
3.1.2	Task description and contributions	51
3.2	Reinforcement learning	52
3.2.1	Policy search approach	53
3.2.2	Value-function approach	54
3.2.3	Model based approach	54
3.3	CEFR and Cambridge Learner's Dictionary	57
3.4	Curriculum Q-Learning	58

3.5	Student response	60
3.6	Visual vocabulary instructional system	60
3.7	Experiments	62
3.7.1	Simulated Students	64
3.7.1.1	Baseline Tutoring Policy	66
3.7.2	Results	66
3.8	Discussion	69
4	Student modelling	73
4.1	Introduction	73
4.2	Related work	77
4.3	Write & Improve	80
4.4	Learning student and task representations	82
4.4.1	Write & Improve data	82
4.4.2	Model architecture	82
4.4.3	Feature set	84
4.4.3.1	Answer embedding	85
4.4.3.2	Question embedding	85
4.4.3.3	Metric embedding	85
4.4.3.4	Grammar error embedding	86
4.4.4	Mean score baseline	87
4.4.5	Logistic regression baseline	87
4.4.6	Evaluation	89
4.4.6.1	Evaluation of score predictions	89
4.4.6.2	Evaluation of grammar embedding predictions	89
4.4.7	Implementation	90
4.5	Results	92
4.6	Discussion	95
4.7	Conclusion	99

5	Human Forgetting Curves	101
5.1	Introduction	101
5.2	Method	103
5.2.1	Duolingo Spaced Repetition Dataset	103
5.2.2	Half-Life Regression (HLR)	104
5.2.3	HLR with Linguistic/Psychological Features (HLR+)	105
5.2.4	Complexity-based Half-Life Regression (C-HLR+)	106
5.2.5	Neural Half-Life Regression (N-HLR+)	107
5.2.6	Evaluation and Implementation	108
5.3	Results and Discussion	108
5.4	Conclusion	111
6	Discussion and Conclusion	115
	Bibliography	121
A	ERRANT Error Types	141

Chapter 1

Introduction

“Let the main object [be]... to seek
and to find a method of instruction,
by which teachers may teach less, but
learners learn more.”

John Amos Comenius

Since the times of Plato and the sophists in ancient Greece, educators have strived to define ideal teaching practices (Saettler, 2004). They studied the human condition to uncover the methods of instruction that can enable teachers to impart as much knowledge as possible, to as many learners as possible, and as efficiently as possible. Many centuries later, researchers, policy makers, technologists and educators continue to revisit our understanding of learning and ideal teaching practices.

It is generally accepted within the educational domain, that the best way to improve learning outcomes of a student is to provide better teaching (Khan, 2020). However, what constitutes “good teaching” has been constantly changing. For years, critics in government have cited inadequate teaching practices as a core reason for the failure in their country’s progress. This usually results in educational reforms, curriculum redesign, and revised teaching practices that are revisited again a decade or two later. For instance, in 1957, when the Soviet Union launched the space satellite “Sputnik”, politicians and pundits criticised the educational curriculum in the United States, citing various reason for why it

was vastly inferior to the Soviet Union. As a result, various educational reforms regarding teaching practices and curricula were made (Conant, 1963). A similar story was repeated in 1983, when a report, *A Nation At Risk*, was published by the National Commission of Excellence in Education (NCEE) highlighting the dismal performance of state schools in the United States compared to other countries. Reforms to teaching practices were made once again (NCEE, 1983).

Whilst it may not be clear which teaching practices achieve the best learning outcomes in each circumstance, what is clear is that good teaching involves an accurate understanding of the student’s current knowledge state. A clear view of the student’s capability enables teachers to make more informed decisions on how to guide the learner through the domain. The teacher’s selection of activities that support the learning process is driven by how they perceive each activity will impact the student’s knowledge state. But maintaining a reliable representation of the student’s understanding is no simple task. In fact, as the size of a class grows, the teacher’s estimation of each student’s knowledge and thus how to determine the optimal sequence of activities suffers (Bloom, 1984). Benjamin Bloom introduced this problem as *The 2 Sigma Problem*, which outlines the difference in performance between students who are tutored one-to-one compared to students in a group setting. By and large, one-to-one tutoring results in the best performing students¹.

However, in reality it is not possible to maintain a one-to-one teacher to pupil ratio, especially with a growing global population and demand for teachers. Therefore, schools must rely on traditional group teaching. But one of the downsides of this form of teaching is that learners on either end of the performance spectrum are under-served (Campbell et al., 2007). So how can we scale the benefits of one-to-one tutoring to a traditional classroom or even at home, where students work without teachers? This is the question that has dominated the discourse for many researchers in the education domain, whether they be policy experts or technologists. Furthermore, it has led to emergence and proliferation of personalised learning, an educational approach which many researchers believe is the key to unlocking the positive effects of one-to-one tutoring in a traditional classroom.

¹(VanLehn, 2011) suggests that the findings in Bloom’s study were overstated.

Furthermore, the notion of personalised learning is considered a sort of holy grail for education and learning. Personalised learning, sometimes referred to as individualised learning, is where the teaching practices of the teacher (whether human or machine) are adapted to each individual learner's needs.

The history of personalised learning is unclear. Depending on how it is defined it can date back to Genevan philosopher Jean-Jacques Rousseau or even further back to the tutor of Alexander the Great, Aristotle (Watters, 2017). However, regardless of which era it originated, in recent years technology has long played a pivotal role in supporting the implementation of personalised learning.

In the 20th century, with the advent of personal computers, researchers have looked towards software and artificial intelligence in order to drive personalised learning. Some of the earliest research published on the use of computers for personalised learning was done under the label, intelligent computer assisted instruction (ICAI). First popularised in the 1970s, ICAI originated from computer aided instruction (CAI) which can be viewed as a traditional way of translating the pedagogical instructions presented by a teacher into program form. In 1982, Sleeman and Brown renamed ICAI to intelligent tutoring systems (ITS) (Sleeman and Brown, 1982).

Much of the early work in ITS was based on rule based systems and heuristics to provide learners with a personalised experience. However, its impact on learning outcomes did not yield the results once promised by the adoption of personal computers in education (Rosé et al., 2019). With the recent rise of machine learning, there has been a resurgence of interest in leveraging newer techniques and methods to achieve better learning outcomes from ITS. There is a plethora of research that explores the use of machine learning for education that includes various aspects of a teacher's role ranging from content creation (Rehm et al., 2020) to instructional sequencing (Bloom, 1968, Corbett and Anderson, 1994, Ritter et al., 2007, Rosen et al., 2018).

Defining all the aspects of what a good teacher does in a classroom or tutoring session is a challenging task. The educational term used to reference a teacher's ability or expertise is *capacity*. And over the years, there has been a constant debate over which teacher

capacities are most relevant and how they should be weighted. However, despite the lack of consensus, teacher capacity can be divided into three broad categories that seem to capture its characterisation over time. These are namely: *knowledge*, *craft skills* and *disposition* (Cochran-Smith et al., 2008). *Knowledge* includes, an understanding of the subject matter, pedagogy, curriculum, and learning theories to name a few. *Craft skills* includes, planning, organising instructional material, managing groups, monitoring and evaluating learning. Finally, *disposition* includes, values and beliefs. This work can be situated within the ‘knowledge’ and ‘craft skills’ capacity groups. However, by no means does the work cover all of either capacity group.

In this thesis, my contributions are focused specifically within one subject area, *language learning*, although the background section does provide a more general overview of the topics covered in this thesis which can be applied to other subject areas as well. A large proportion of the recent work in educational technology is focused around STEM subjects and therefore, in order to fill this gap, the focus of the work presented in this thesis is language learning. Within language learning I deep dive into the task of *personalised instructional sequencing*² and the role of machine learning within this task. Instructional sequencing is how content is presented to a particular student. The output of an instructional sequencing task is a curriculum. One can think of personalised instructional sequencing as a method of constructing a dynamic curriculum that evolves over time with the performance of the student. It is important to note that while personalised instructional sequencing may seem like a very narrow and focused task, the implications of personalised instructional sequencing expand far beyond its immediate function. For example, in order to develop a method of personalised instructional sequencing one has to consider human cognitive functions like forgetting and knowledge representation. In fact, I would argue that instructional sequencing is at the centre of the teaching and learning experience. It is the central processing unit (CPU) of education that decides what, when and how information is delivered. Most other educational technology tasks e.g. automated

²Henceforth, I may use personalised instructional sequencing and instructional sequencing interchangeably

assessment, content creation, can be executed without the function of time. Instructional sequencing introduces not only the complexity of temporality but, in order to be effective, must consider the impact of time on learning which requires some proxy for measuring learning; typically done through automated assessment. Furthermore, the *personalised* in personalised instructional sequencing requires additional considerations to be made including how learners forget and how we might codify their knowledge and the work in this thesis reflects that.

Looking forward, in Chapters 3 to 5, I introduce machine learning methods and approaches to model personalised instructional sequencing while accounting for the aforementioned considerations. While prior work in personalised instructional sequencing relies mainly on heuristics or rule based systems supported by a theory of learning to guide learners through the language curriculum, in this thesis I present methods that not only provide a quantitative approach to instructional sequencing for language learning, but also outline a framework to learn about language learning.

The first study in this thesis presents a reinforcement learning framework that is grounded in theories of language learning to guide a student through a vocabulary learning task. The task presents an image and requires the student to enter the word associated with the image. This work leverages ideas from some popular concepts in educational psychology such as the zone of proximal development (ZPD). ZPD is viewed as the distance between what a learner can do without help, and what a learner can do with the support of an expert such as a teacher. The model leverages Common European Framework of Reference for Languages (CEFR) levels to categorise each vocabulary word. CEFR levels range from A1 to C2, where C2 is the most difficult level³. The zone where the student can learn the word without help is the student's current CEFR level. At each point in time the learner is at some approximate CEFR level and the teacher must make a decision on which items to present next. It turns out that reinforcement learning captures the essence of this scenario quite effectively, whereby the student's knowledge is the current CEFR state and the teacher's range of possible items or vocabulary words to present are

³CEFR levels are as follows: A1, A2, B1, B2, C1 and C2

the actions. One can consider the process of learning a policy through the reinforcement learning model as instructional sequencing. The personalised component of instructional sequencing is derived from unique knowledge states for each student driving the optimal actions and thus policy. This will be explained in more detail in Chapter 3.

The true value of a policy derived from a reinforcement learning model is highly dependent on the quality of the state representation (Mnih et al., 2013) or student knowledge states, in the task of personalised instructional sequencing. As a parallel, the value of a teacher’s actions for a particular student in the classroom is highly dependent on how well the teacher understands the student’s strengths and weaknesses. If the understanding is relatively shallow, the teacher’s actions and decisions are likely to be suboptimal. In the first study, the reinforcement learning model learns a policy based on a fairly rudimentary representation of the student’s knowledge state. Therefore, in the second study I explore methods of improving the quality of these representations.

Motivated by the recent success of deep learning in extracting high-level features, in Chapter 4, I investigate how the properties of deep learning can facilitate training improved student state representations for instructional sequencing. In this work, I leverage collaborative filtering; a technique that is popular in recommendation systems. It is grounded in the idea that people who are similar will likely enjoy similar products or movies. I extend that idea to student’s who have similar scores on a common set of tasks, will likely have similar understanding and misconceptions within the content. Using this framework, I aim to predict the performance of a student on tasks that he or she has never attempted before in order to determine whether those tasks are appropriate, or within the ZPD.

Using this approach, I develop an encoder that takes in the student’s data and constructs a student representation. The representations is in turn used for the prediction of student performance across all tasks. Furthermore, as the students are represented as external state vectors (Wexler, 1970), two students who are similar should be closer to each other in vector space. In order to train the system, I leverage data from Write&Improve, an online writing platform that contains prompts and provides instant feedback on the

student’s writing with a score. In addition to presenting a method of developing student representations for language learning, I also present an example of how these techniques can be used to uncover learning patterns in students (learn about learning). Specifically, in this work, I analyse the distribution of grammatical errors across different levels of student to identify whether there are certain error types are distinctive of beginner, intermediate, and advanced language learners.

In addition to knowledge state representation, incorporating the human cognitive phenomenon of forgetting is a critical part of ‘personal’ in personalised instructional sequencing. Without accounting for the effect of time in instructional sequencing, we will fail to converge at an optimal policy or curriculum. Furthermore, without modelling recall or *remembering*, the first block in Bloom’s taxonomy (Bloom et al., 1956), we are missing out on, arguably, the most fundamental process of learning. The reasons for forgetting is still ill understood, but there are several methods to counter the forgetting. One of those methods is known as spaced repetition learning. Spaced repetition is the idea that reviewing items in an increasingly spaced fashion counters the forgetting. In order to optimise spaced repetition learning, we must have an approximation for when a student is about to forget. In 1885, Ebbinghaus conducted an experiment on himself and presented what is known as the forgetting curve. More details about forgetting and spaced repetition learning are discussed in Chapter 2 and Chapter 5. In the third study, I explore and evaluate various approaches to modelling the forgetting curve that adapts according to the content being learned. As my work is in the language learning domain, I incorporate psycholinguistic features such as word complexity to train my models. The study explores whether we can more effectively model forgetting using neural networks, which are known to capture and model latent and hidden features. I also examine the predictability rank of each psycholinguistic feature in predicting whether a student will accurately recall a particular word. For this study, I leverage the publicly available Duolingo dataset for spaced repetition learning in language.

Each of the aforementioned studies look at the benefits of machine learning in personalised instructional sequencing for language learning but also the limitations. While

there is a lot of fundamental research in machine learning methods that argues it is not necessarily important to understand the inner workings of the brain to further research, this thesis suggests that at least when it comes to educational technology, understanding how humans learn is critical in developing better machine learning models for teaching. Furthermore, I also show that machine learning can be used as a tool for ITS but also as a tool for learning about learning or understanding how human learning works. In the first study, I show how a reinforcement learning framework can be considered as an effective method of controlling instructional sequencing. I also show how some of the characteristics of the framework have parallels with different concepts in theories of language acquisition. In the second study, I show how deep learning and collaborative filtering might help us uncover unobserved attributes of learners. Finally, in the third study, I present a state of the art system for modelling forgetting in a vocabulary learning context. I show that word complexity plays a pivotal role in determining how likely the student is to forget the word. Furthermore, I also show that grounding models in certain theories of learning can greatly improve model performance. These findings also make the case for incorporating additional innate bias in the way we use machine learning models in education.

The structure of this thesis is as follows: Chapter 2 introduces instructional sequencing and its origins. It highlights some of the key instructional design theories in education over the centuries that helped motivate my experimental design decisions but that I hope will motivate the design decisions of future teachers, computer scientists and researchers in the educational domain. This chapter also presents a background of knowledge representation which is a critical component of personalised instructional sequencing. Finally, I present a brief overview of the human phenomenon of forgetting in the context of learning. In Chapter 3 I present a new approach to leverage reinforcement learning for visual vocabulary learning which has been adapted from a published paper at the ViGIL workshop at NeurIPS (Zaidi et al., 2017). In this chapter, I also discuss the parallels between the reinforcement learning models and theories of instructional design and learning. In Chapter 4, I investigate knowledge representation, the student model and its role in instructional sequencing. Furthermore, Chapter 4 presents a novel neural network

inspired by collaborative filtering models to develop representations of students language understanding. These representations can be leveraged as the states of a reinforcement learning model to converge at an optimal policy or personalised instructional sequence. This chapter has been adapted from my paper published at the EDM conference (Zaidi et al., 2019). In Chapter 5, I explore the role of forgetting within instructional sequencing. The work explores previous methods of modelling forgetting. These models can then be used to evaluate optimal spacing for a spaced repetition technique. I present a new method using neural networks to adaptively model the forgetting curve which is conditioned on the student’s prior performance and the particular word or words they are learning. The model also highlights the importance of certain psycholinguistic features, specifically word complexity, in predicting recall. This chapter has been adapted from a conference paper published at AIED (Zaidi et al., 2020). Chapter 6 discusses the key findings of the thesis and identifies key future works to be explored.

Since this work is about personalised instructional sequencing, I felt it was necessary to provide a map and overview of how to navigate the thesis. The topical navigation section provides a quick way for interested readers to find information relevant to them. Although there are inter-chapter references, I have tried my best to keep the chapters as self-encompassing as possible while still retaining an overall narrative for the thesis. The experimental chapter summary is for readers who want to get a gist for what experiments I ran, the methods I used, and the main contributions of those chapters for the educational data mining community.

1.1 Topical navigation

Use the flow diagrams below in order to navigate the thesis based on topical interests

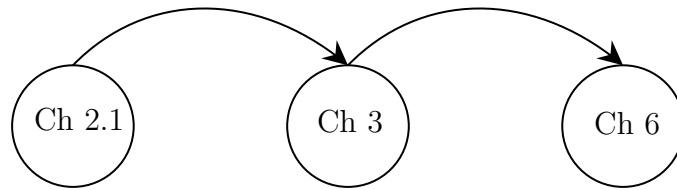


Figure 1.1: Personalised instructional sequencing history and methods

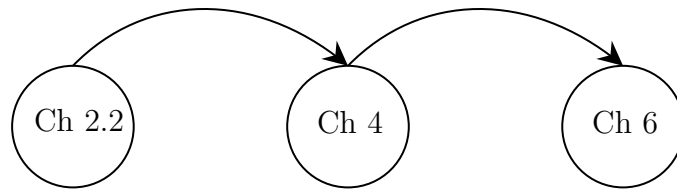


Figure 1.2: Student knowledge representation challenges and approaches

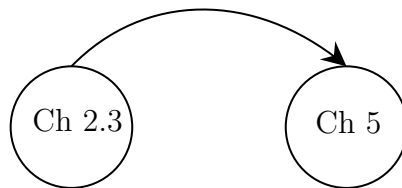


Figure 1.3: Human forgetting curves and psychologically driven model selection

1.2 Experimental chapter summary

The table below provides a summary of the experimental chapters of this thesis with methods used and main contributions.

Ch	Topic	Method	Contribution
3	Personalised instructional sequencing for language learning	Reinforcement learning; convolutional neural networks; word embeddings	Illustrated the importance of leveraging domain structure for model design; designed an educationally motivated reward function; presented a reinforcement learning personalised instructional sequencing system that requires minimal teacher intervention
4	Student knowledge representation	Neural collaborative filtering; word embeddings; auxiliary objective functions	Novel model for developing student knowledge representation using neural collaborative filtering approach; predicting student performance on unseen tasks by learning from similar students; leveraging grammar error distributions to identify what errors students of different proficiencies make
5	Adaptive forgetting curves for language learning	Neural networks ; word complexity; psychologically driven model selection	State of the art model for predicting probability of recall of vocabulary words; identifying word complexity as a predictive signal for recall probability; illustrating how leveraging psychological theories to drive model selection results in better performance than naive neural network;

Chapter 2

Background

Instructional sequencing has an extensive history that has been influenced by centuries of thinking, experimentation and more recently, data. Although in this thesis I do my best to talk about instructional sequencing in isolation, ultimately, it is difficult to talk about instructional sequencing without talking about knowledge versus information and discussing the overall objectives of teaching and learning.

2.1 Instructional Sequencing

The origins and evolution of instructional sequencing is an important section as it will deeply motivate the design decisions I make in the rest of the chapters. It is my hope that this chapter will become an important contribution to the literature on instructional sequencing and personalised learning but also enable future researchers to consider the work that has come before. However, this is by no means a fully exhaustive review of the history of instructional sequencing but instead an attempt to summarise what I consider some of the key milestones through the times.

2.1.1 Origins of instructional sequencing

Instructional sequencing is the order by which a set of materials are presented to a learner. The effect of instructional sequencing is often defined as the performance improvements

realised when the same set of materials are presented to a learner in a different order (Ritter et al., 2007). The performance improvements serve as a proxy for knowledge acquisition but since knowledge cannot be probed directly, it is common practice to rely on test scores or communicable success in the context of language learning to obtain that signal.

The history of instructional sequencing can be dated back to the times of the ancient Greeks. The Sophists (4-5 B.C.), or teachers in ancient Greece, were the first in recorded history to implement the method of mass instruction. Through this, they were able to develop and refine a methodology of presenting instruction effectively (Ritter et al., 2007). The Sophists systematised instruction by starting with the presentation of the rules of writing and speaking. Thereafter, they showed examples which were copied and emulated by their pupils. Finally, students were expected to apply the learned rules in a different context.

However, the teachings of the early Sophists were largely overwritten by the rise of Plato and his successors. In particular, while the Sophists believed that all men were capable of intelligent, socially responsible self-rule, Plato believed that you were either destined for low or high society and that it was ultimately only the high society or the rich are capable of virtue (Saettler, 2004). Virtue was a critical part of education, as it still remains today, mostly in the early years of education.

During the times of the ancient Greeks, presenting instruction was seen as a vital art form that civilised society. It was considered an art form because it was understood that there is a fundamental difference between simply presenting instruction and presenting instruction in such a way that translated information into knowledge. Plato and his successors, being philosophers, understood the nature of knowledge and sought to ensure that this was reflected in their teaching practices.

Many centuries later, in the early 1600s, Johann Amos Comenius (1592-1670), a Czech philosopher, proposed in his book, *Didactica Magna* (The Great Didactic), that the order of instruction should follow the natural development of a learner. He stated that instruction should be designed for age, interest and capacity of the learner. Comenius was instrumental

in shaping modern personalised instructional sequencing (Piaget, 1993). Furthermore, he called for the use of illustration in instruction as well as stating the importance of organising content from simple to difficult. For language learning, he promoted the practice of teaching both writing and reading together and claimed it was irrational to learn a foreign language before the native language has been learned. Unfortunately, for many centuries, Comenius' work in instructional methods was largely unknown and only rediscovered in the mid-19th century.

In the early 1800s, Joseph Lancaster (1778-1838) introduced the Lancasterian Monitorial instruction. Lancaster was one of the first to formalise classroom organisation, develop a graded plan for group instruction and incorporate educational economics. One of the main limitations of one-to-one tutoring, as mentioned in the introduction, is its cost. In 1819, Lancaster had several schools in Philadelphia where the pupil to teacher ratio, in some cases, was 1 to 284. This was done to keep the cost low. In this system, personalisation was incorporated by the teacher training up to 50 pupils. Each of these pupils would then play the role of group monitor (modern day teacher assistant) for 10 additional pupils. This ensured that in the context of a large classroom, there was still a level of personalisation provided through the monitor.

The content was presented from simple to complex. The monitor would introduce the rules, show an example on a board and erase it. Thereafter, each pupil was required to solve the same example. This is the first known example of how instruction presentation was scaled across large groups of students with some element, albeit at a minimal level, of personalisation. Unfortunately, after some initial success with this model, Lancaster received a lot of criticism for his alleged poor standards and harsh discipline and as a result his teaching methods were discredited soon after.

Johann Henrich Pestalozzi (1746-1827) was one of the first to incorporate psychology into instructional sequence and methods. Influenced by the teachings of the Genevan philosopher Jean Jacques Rousseau (1712-1778), Pestalozzi set up experimental schools where he applied different methods of teaching. While he did not use empirical methods, he did predict that in the future an emergence of *instructional science* would occur. In

his teachings he emphasised that every learner has different abilities and therefore the teaching must be adapted to the learner. He also promoted that instructional sequence should follow the natural development process of the learner. For example, he taught language by starting with sounds then syllables, words and finally sentences. In early 1800s, Johann Fredrich Herbart (1776-1841) built on the teachings of Pestalozzi and Comenius to show how instructors can and should link prerequisite knowledge to future teaching content. He claimed that the main task of instruction is to ensure the proper sequence and connection of ideas. He proposed that the sequence of instruction should observe the following structure (Saettler, 2004):

1. Clearness - reduce a concept to its elemental components and teach those concepts in isolation.
2. Association - connect different concepts together to show how they're related
3. System - to study the system as a whole in order to understand the role of each component and its relative importance
4. Method - once a system and its components are understood, apply the system to a new problem.

2.1.2 Emergence of scientific instructional sequencing

Until now, all of the innovation in instructional methods was developed, refined and evaluated based on observation. The methods were not backed by an empirical scientific process. That changed in the early 1900s with the work of Edward Thorndike (1874-1949). Thorndike, an educational psychologist, had developed the first scientific theory of learning, *connectionism theory*. Connectionism theory states that learning is a result of practice and associations or connections formed between stimuli and responses (reward system). Thorndike was particularly interested in the application of his findings in education including spelling and reading (Thorndike and Lorge, 1944), measuring intelligence (Thorndike et al., 1926), and adult learning (Thorndike et al., 1928). Based on the early works of

Hermann Ebbinghaus (1850-1909), Thorndike developed the decay theory, a theory that states memory fades with the passage of time (Thorndike, 1913). This theory was highly criticised by McGeoch with his interference theory (McGeoch, 1932). Interference theory states that forgetting occurs because memories interfere with one another. The evidence, according to Berman et al. (2009), supports interference-related decay over temporal decay.

Thorndike's learning theory played a critical role in the development of operant conditioning within behaviourism. Operant conditioning is the idea of reward and punishments to promote desired behaviour. Whilst Thorndike was credited with discovering operant conditioning, B.F. Skinner (Skinner, 1938) was credited with coining the term and formalising it. The concept of operant conditioning still plays an important role in personalised instructional sequencing, specifically when deciding on how and when to provide feedback before proceeding to the next task.

In the same era, Maria Montessori (1870-1952), a medical doctor by training, also rose to prominence through her publication *Scientific Pedagogy as Applied to Child Education in Children's Houses*. Although Montessori's instructional principles were not based on statistical design and evaluation, she did leverage her clinical observations to devise optimal strategies for learning (Montessori, 1912). It was these observations that ultimately led to her discovery of the importance of sensory learning. *Sensory learning* is the notion that children should use their senses when they explore and learn.

Montessori developed a methodology that was grounded in instructional adaptivity and promoted two key tenets: individuality of the student and freedom of direction. Her methods relied on anticipating what the learner was trying to do and developing a bespoke and individualised plan to guide the learner through that experience. It also ensured the teacher was not overly dominant in the teacher-student relationship. Montessori's success marked, in many ways, the beginning of the rise of individualisation of instruction.

In 1912, Frederic Burk (1862-1924) developed the first *formal system* of individualised instruction. Burk, with the support of his colleagues, reordered the course material to meet the individual needs of each learner. The materials were ordered in such a way that ensured teacher intervention was kept to a minimum. Similar work was done by

Carleton Washburn (1889-1968), when he developed the Winnetka Plan in 1919. This plan not only allowed learners to proceed at different rates, but also at different rates for different subjects. In the same year in Dalton, Massachusetts, the Dalton Plan was developed by Helen Parkhurst (1887-1973). This plan was designed for students with physical disabilities. In this plan, students were given a certain number of teacher hours for each assignment. The students were free to use the hours in whichever way they required to complete the assignment. Similar to the Winnetka Plan, the Dalton Plan promoted individualised instruction and self-learning. Between 1925-1935, Henry Clinton Morrison (1871-1945) proposed the highly influential Morrison Plan. In his plan or method, he identified a five-step process for developing personalised instruction (Morrison, 1926):

1. Pretest
2. Teaching
3. Testing the result of instruction
4. Changing instruction procedure
5. Repeat (teaching and testing again) until mastery is achieved

Morrison distinguished between learning, performance and adaptation. Mastery learning is when the student has achieved a grasp of the subject matter. The ability to apply that subject matter or skill is performance. Adaption is the process of applying that skill in different situations. This is an important distinction in instructional sequencing especially when we consider what it is we want the students to ultimately be capable of. From Morrison's insight, we can deduce that our instructional objectives should adapt over time. That is, in the beginning, we should aim for the student to understand the subject matter. Thereafter, we proceed to supporting the student in applying that subject matter in a constrained environment. Finally, we enable the transfer of those skills and capabilities in different environments and situations. This idea was built upon many years later in the well known Bloom's Taxonomy (Bloom et al., 1956).

In the late 1930s, B.F. Skinner (1904-1990), a professor at Harvard University, began his journey to understand the science of behaviour. Among his many interests, he was particularly curious about verbal learning, teaching machines, and the influence of reinforcement on behaviours. Skinner believed that one of the main issues with traditional learning was the lack of consistent reinforcement for desired behaviour. Furthermore, he believed the role of the teacher was to architect instruction and teaching in such a way that maximises reinforcement. This is where his interest in teaching machines came into play. The role of Skinner and his behaviourist perspective played a critical role in helping to define personalised instructional sequencing methods. This is especially relevant when it came to discussing how the order of topics and subsequent reinforcement signal affects long term retention. Skinner's theories have shaped many aspects of education. These influences can be attributed to his work in *The Technology of Teaching*. The application of Skinner's theories can be seen in the Keller Plan (Keller, 1967) and Lindsley's precision teaching system (Lindsley, 1991).

2.1.3 Computational instructional sequencing

In the 1950s, the advent of computers led to the rise of Computer Assisted Instruction (CAI). Much of the early work of CAI was conducted by IBM although other prominent names in the field included the likes of Gordon Pask (1928-1996) who developed adaptive machines, and Omar Khayyam Moore (1920-2006), who introduced autotelic responsive environments to teach children how to read. Most CAI implementations were inspired by Skinner's behaviourist approaches. The early works in CAI were based on the drill-and-practice approach. This is where students were provided with a question and a possible selection of answers. If the answer was wrong, the screen would flash, "WRONG". Alternatively, if the answer was correct, it would proceed to the next question. Furthermore, the curriculum was controlled by the teachers and allowed very little flexibility on the part of the student.

In 1967, Richard Atkinson and Patrick Suppes (1922-2014), two highly influential thinkers in the CAI domain, setup the Computer Curriculum Corporation (CCC). Atkinson

in particular, has had a deep interest in language learning including areas like second language acquisition (Atkinson, 1975), word recognition (Atkinson and Juola, 1973), mnemonic methods for vocabulary acquisition (Atkinson and Raugh, 1975). In CCC, Atkinson and Suppes developed CAI drill-and-practice materials for mathematics, programming, language and reading. Although it showed great improvements on the Stanford Achievement Test (SAT) and computational skills, the impact on reading and language arts did not yield a similar improvement. This is an important observation, especially given the focus of this thesis. It supports the assumption that the instructional sequence methodology for language may well differ from the approach to mathematics. By the mid 1970s, despite its initial promise and extensive government funding, it was clear that CAI had not succeeded in delivering the impact it had desired. The failure of CAI led to a new research direction towards cognitive psychology, instead of behaviourism, to drive educational technology.

One of the main critics of Skinner's behaviourist approach was Seymour Papert (1928-2016) who propagated the views of his supervisor and mentor, Jean Piaget (1896-1980) and his *constructionist theory* of learning. This theory stated that the best way to learn was by building and exploring in the real world. Under this view, when students learn, they develop mental models of the world. Therefore when they learn by exploring, students can apply their existing knowledge and understanding of the world to acquire new knowledge. The constructionist approach is personalised and learner-centric where the role of the teacher shifts from a lecturer and provider of information to a coach and guide. In 1967, Papert developed LOGO, an educational programming language while he was at MIT. LOGO was designed to be an environment that encourages exploration. Papert went on to found the Epistemology and Learning Research Group at MIT which was influential in starting the MIT Media Lab.

In the 1960s several theories and models of instruction emerged. In 1966, Jerome Bruner (1915-2016), a prominent American psychologist, in his book *Towards a Theory of Instruction* (Bruner et al., 1966), proposed a theory of instruction that included the following four criteria:

1. Predisposition to learn. The student must be willing to learn. This can be done by creating a level of uncertainty in the learning process that allows the student to explore.
2. Structure of knowledge. The curriculum should be structured in such a way that the body of knowledge should be easily accessible to students.
3. Sequencing. The body of knowledge should be presented in a sequence that makes concepts easier to grasp, transfer and apply. Bruner suggested the following order: enactive (hand-on, concrete), iconic (images and visuals), symbolic (logical and mathematical expressions).
4. Reinforcement. Bruner stated that the nature and pacing of rewards and punishments should be specified. Rewards should move from extrinsic, such as teacher praise, to intrinsic, such as the ability to solve a problem.

Although Bruner's work and theories reside within the cognitive approaches to instructional design, there are some elements of behaviourism namely the emphasis on reinforcement.

In 1956 Robert Gagné (1916-2002), a former air force pilot trainer, came up with 8 different ways to learn. The ways to learn are organised by increasing complexity (Gagné et al., 1985).

1. signal learning - this is seen as the simplest form of learning and consists of classical conditioning as discussed by Pavlov (Pavlov, 1910). An example of this is the salivation of a dog upon hearing the sound of food being poured into a metal dish. The signal is the sound of food in the dish and the conditioned response is salivation.
2. stimulus response learning - this form of learning is similar to reinforcement learning proposed by Skinner. This is where the student is either given an award or punishment after each response. This type of learning is fairly precise in comparison to signal learning that is diffuse and emotional.

3. chaining - This is where the student can connect two or more previously learnt stimulus-response bonds. An example of this is riding a bike or playing the piano, which are both examples of activities that require hand-eye coordination.
4. verbal association - links between items that are verbal in nature. This is key for language development. Verbal association is a type of chaining with an observation is linked with a verbal response. For example, when a child says "ball" upon observing one.
5. discrimination learning - this is the ability to provide different responses to different stimuli. This results in what is known as interference, which is thought to be one of the main causes of forgetting. An example of this learning might be in a classroom, where a teacher attempts to call on each student by his/her correct name.
6. concept learning - this is the ability to form a consistent response to stimuli and categorise it into a class. This enables students to generalise concepts. An example of this might be the ability to correctly identify whether an image is of a cat while presenting the student with different species of cats. Even though the presented images of cats vary in their physical appearance, they have some shared abstract features that help the student develop a concept of a cat.
7. rule learning - this is the process by which students understand the relationship between concepts and are able to apply those concepts in situations they have not previously encountered. For example a student might observe a kitten making a "meow" sound to its mother and learn a rule that the "meow" sound from a kitten usually indicates to the mother cat that the kitten is hungry or cold.
8. problem solving - this is considered the highest level of learning which is the ability to create a rule, algorithm or solution to solve a particular problem. An example of might be a question on a physics exam where the student is asked to determine which vehicle will reach the destination first given varying conditions for the vehicle speed and the route. This requires the student to chain various rules that he/she

has learned to calculate the relevant values. The student must then come up with a new rule that compares these values to reach an answer.

Gagné's 8 different ways of learning presents a valuable framework to consider when designing instructional design tools.

Another approach to computational instructional sequencing was introduced by Lev Landa (1927-1999). He proposed the algo-heuristic theory of instruction (Landa, 1983). This theory presents a new framework to combine the best of algorithmic instructional sequencing and heuristic approaches. While algorithms are precise and measurable there are some tasks which cannot be clearly defined algorithmically because their underlying principles are ill understood. For those tasks, heuristics are a better option. Once it is identified whether each concept is better presented algorithmically or heuristically, they can be organised into a system of instruction that is learner led, combining both approaches.

2.1.4 Artificial intelligence and instructional sequencing

In the 1970s a new term, intelligent computer assisted instruction (ICAI) signalled a new wave of tutoring systems that aimed to automatically adapt to the learner. It was believed that ICAI would change the feasibility of personalised instruction, creating the ability to scale it across the population. Some well known ICAI systems include SOPHIE (Brown et al., 1975) which was designed as a tutorial dialogue system for question-answering. It focuses on allowing students to have a reactive learning environment in which to explore their ideas. The system employed very basic natural language techniques to provide a response to the student. The first domain used for SOPHIE was electronic troubleshooting. Another popular system was BUGGY (Brown and Burton, 1978). It was inspired by the idea of a computer "bug". In the context of a student, a "bug" was the student's misconception about a concept. Previous models relied on models that capture what the student understood. However in BUGGY, misconceptions are explicitly represented in the framework. The initial domain of choice for BUGGY was arithmetic skill and it specifically became well known for place-value subtraction.

Alongside the popularisation of the perceptron, in 1960 Ronald Howard published a book by the name of *Dynamic Programming and Markov Decision Processes (MDP)* (Howard, 1960a) as well as an article called “Machine-Aided Learning” (Howard, 1960b). The book went on to become a foundational contribution to reinforcement learning. The article, on the other hand, discussed the potential for computers to provide individualised instructional sequencing. Richard Smallwood, Howard’s doctoral student built upon Howard’s work in this dissertation “A Decision Structure for Teaching Machines” (Smallwood, 1962). The dissertation presented an approach to adaptively ordered instruction using MDPs. This made instructional sequencing one of the earliest use cases for MDPs.

Building on the work of Howard and Smallwood, in 1972, Richard Atkinson introduced the four criteria or ingredients that must be satisfied in order to reach the optimal instructional strategy (Atkinson, 1972a). Atkinson then pointed out the duality between MDPs and his four ingredients for optional instructional strategy. The four criteria are summarised as follows:

1. A model of the learning process of the student
2. A range of possible instructional actions
3. Identification of key educational and instructional objectives
4. A method of evaluation that enables “cost” to be assigned to each instructional action in relation to the instructional objectives.

Atkinson then embeds each of the four points above in the MDP paradigm. Let’s first review the components of the MDP.

MDPs consist of: states S , actions A , transition probabilities of going from a given state s to an alternative state s' due to an action a is represented in the form $T(s'|s, a)$, reward for taking a particular action a in a given state s represented as $R(s|a)$ and H which is the number of time steps.

Atkinson presents instructional sequencing in the form of an MDP by mapping states S to the student’s current knowledge state, the set of actions A are the possible learning

activities that can affect the state s , and the reward R is the overall learning gains observed by the student. Atkinson also draws parallels between the transition function $T(s'|s, a)$ and the learning process. That is, the probability of going from one state to another can be grounded in theory of learning. Presenting personalised instructional sequencing in an MDP framework opens up a range of powerful tools and techniques that can be used to tackle the instructional sequencing task. Some of these techniques are discussed in Chapter 3.

2.2 Knowledge representation

Knowledge representation is a fundamental part of artificial intelligence. It is the way information is represented in a computational environment and it is these representations that are ultimately processed and evaluated by a computer to make decisions (Brachman and Levesque, 2004). Therefore, the quality of representations directly impact the quality of decision making.

In an educational setting, knowledge representation can refer to the cognitive state of the student, the question being presented by the tutor, the answer being provided by the student. Therefore, the quality of these representations can be considered a critical or rate-limiting step in the overall quality of intelligent tutoring systems. In a personalised instructional sequencing system, the representation of the student and curriculum is the signal that determines what, when and how content is presented. Representations in these system can take many forms, from simple numeric forms to more complex networks (Brusilovsky, 1994, Elliott, 1993).

In 1974, John Self published a paper titled “Student models in computer-aided instruction”. Self (1974) states that the knowledge related to teaching performance can be divided into three categories:

- (a) knowledge of how to teach (which includes knowledge of students in general)
- (b) knowledge of what is being taught

(c) knowledge of who is being taught (i.e. knowledge of one student in particular)

This section is specifically concerned with (c) which, according to Self (1974), is the most relevant category for individualised instruction. Capturing what a student understands as well as his/her misconceptions are critical to designing personalised instruction.

Typically, a teacher will have a mental model of each student's capabilities and knowledge. It is this mental model that needs to be explicitly represented in an intelligent tutoring system. Peplinski (1970) approached this task by counting the number of questions and correct answers provided by the student. Using the ratio of correct answers to questions, the system automatically determines whether to increase or decrease the difficulty of the questions.

Carbonell (1970) represents knowledge as a semantic network and uses temporary tags to indicate whether a particular node of the network has been asked to the student and whether it was answered correctly. Each node represents a concept or question. The collection of nodes and temporary tags then represent the student's current knowledge state. This approach was also employed by Wexler (1970) who used it not only to organise knowledge but also to generate questions to test the knowledge of students. Brown et al. (1972) uses an external state vector to represent student knowledge. In the vector, each dimension represents a concept and the value within the dimension acts as a flag that is modified based on the system's interaction with the student. That flag can represent anything from indicating whether the question has been asked to how likely it is that the student understands this concept.

Whether representing knowledge as a semantic network or an external state vector, there is still an important consideration to make regarding how you will measure the student's knowledge to the target or domain knowledge. More specifically, is student knowledge representation a subset of the domain knowledge representation or is it its own unique representation with different nodes and dimensions?

A model where the student knowledge is a subset of the domain model is known as an overlay model (Carr and Goldstein, 1977). Whilst it provides an elegant yet simple solution to monitoring students' knowledge states and comparing it to the domain

knowledge representation, it has some limitations. For instance, the student might have certain skills that are not explicitly represented in the domain model which may still provide some transferability in the domain being studied. Additionally, capturing misconceptions in an overlay model is challenging as there is no list of predefined misconceptions that cover all possible reasons why a student has not understood the concept within a particular domain. Finally, there is a question of how to divide and group the concepts within the domain which will then serve as nodes or dimensions in network and external state vector representations, respectively. This is not only a resource intensive task, there is also likely to be significant difference of division from one curriculum designer to another.

One possible method of addressing the challenge of dividing up the domain into different concepts is to develop a domain model that is an output of numerous experts' division of the domain. Diederich et al. (1961) achieved this by asking a large number of admission panellists to identify errors on the same set of student writing submissions for college admission. Those errors were mapped onto an error grid which then provided writing teachers with a specific profile for each student. Since the division and taxonomy of language errors is highly varied, constructing an error grid that is an amalgamation of error notions from several writing teachers is a valuable alternative. It also provides a standardised yet personalised way for representing each student's ability.

Another aspect to consider for student representation is bandwidth. Bandwidth is the amount of information available about the student knowledge representation over time. That is, how does the student knowledge representation change with exposure to new instructional material? For example, programming tutor PROUST (Soloway and Johnson, 1984) only captures the final completed code from the students. This might be efficient from a computational memory perspective, but can create challenges if the educator is trying to understand the root cause of potential knowledge gaps and misconceptions. In contrast, step-wise tutoring systems such as Cardiac Tutor (Eliot et al., 1996) stored and evaluated each step of the student's action. The Cardiac Tutor provides students with an environment for which to practise advanced cardiac life support (ACLS). Due to higher bandwidth, the tutor is able to better diagnose the student's errors and misconceptions by

providing a knowledge path that can be debugged.

As mentioned earlier in the chapter, knowledge representations are important because they are the input that intelligent tutoring systems then use to make decisions. Therefore, any decision regarding the student's knowledge but also the difficulty of the question and the validity of the student response must be derived from the knowledge representations. We see an example of capturing question difficulty explicitly in the knowledge representations in AnimalWatch. AnimalWatch was a tutoring system that provided students with mathematical instruction for addition, subtraction, fractions, decimals, and percentages. It was targeted at students aged 10 to 12 (Beal and Arroyo, 2002). AnimalWatch was a generative tutoring system that automatically adapted to the students' learning style and only moves on to the next topic if mastery of a topic is achieved. The domain model was arranged as a network of nodes where each node represented a skill. The difficulty of the question was determined explicitly by adding the number of skills (nodes) required to solve the question. The greater the number of skills, the harder the problem. The student model in AnimalWatch captured how long the student took to give a response, how the student reacted to different hints, and incorrect responses. The student model was designed as an overlay of the domain model.

All the models described above are deterministic, where the student either has learned that particular skill or has not. Pump Algebra Tutor (PAT) provided a more probabilistic approach to knowledge representation that was grounded in cognitive models. PAT was a cognitive tutor for algebraic problem solving. This tutoring system, similar to the Cardiac Tutor (Eliot et al., 1996) modelled the entire path the student took to get to the solution (Koedinger and Anderson, 1998, Koedinger and Sueker, 1996, Koedinger et al., 1997) or employed a high bandwidth approach. PAT was based on a theory of human cognition called ACT-R (Lebiere and Anderson, 1993). ACT-R divides knowledge into two categories, declarative and procedural. Declarative knowledge consists of units called chunks and are generally facts or concepts discussed in written form. Procedural knowledge is represented by procedural rules and consists of if-then statements. According to the ACT-R model, procedural knowledge cannot be learned by sitting in a lecture and can only be learned

by “doing” e.g. driving a car, swimming. The PAT model estimates the likelihood that a procedural rule or declarative chunk is understood by a student. Based on this estimation, the tutor adapts its instruction and is able to prioritise the order of instruction.

The study of knowledge representations cannot simply be restricted to just its structural representation. Another aspect to consider is when, how and to what extent the student representation will be modified by the insertion or addition of new knowledge and material i.e. how do we adapt the student’s representation as the student acquires new knowledge. The Knowledge Learning and Instruction (KLI) framework (Koedinger et al., 2010) addresses this challenge in some respect by presenting a cognitively motivated framework for understanding the relationship instruction, assessment and learning. The KLI framework consists of three types of events, instructional events (IE), assessment events (AE), and learning events (LE). IEs are activities that are presented by a teacher or tutor (human or machine). AEs include activities such as taking exams. Some AEs are also IEs, for example teacher-student discussions which can serve as a medium of instruction but also a form of assessment. IEs are what cause LEs. LEs are unobservable processes which the authors divide into three different classes: 1) memory and fluency, 2) induction and refinement, and 3) understanding and sense making. LEs cause unobservable changes to the knowledge components (KCs). The authors define KCs as “an acquired unit of cognitive function or structure that can be inferred from performance on a set of related tasks”. As KCs are unobservable directly, they can only be measured or gauged through AEs.

The KLI framework also discusses challenges and approaches of defining KCs. The challenge arises from the fact that different curricula experts have varied perspectives on how to divide the curriculum into components.

A group of KCs can form a *knowledge state* for a particular subject. A group of knowledge states can form a *knowledge space*. The idea of a knowledge space allows us to view knowledge beyond the confines of a particular subject/domain. This is beneficial especially in subjects or specific skills where transferability is inherent. For example, if an English speaking student is learning French, there might be some complex or higher

level words that sound very similar to English words and therefore, the student domain knowledge that would not necessarily correlate with his/her general language level.

Falmagne et al. (1990) explores how knowledge spaces can be designed probabilistically. The author presents a mathematical framework that enables the discovery of how different knowledge states relate to each other in a knowledge space. Falmagne et al. (1990)'s system would be able to predict how likely it is that a student knows who the United States Vice President is given he/she knows 5 of the 7 United States Supreme Court Justices. The knowledge spaces are initially designed by an expert in the subject but are refined by empirical data captured from tutor-student interaction.

So far I have outlined the importance of knowledge representations and some of the challenges, including:

- Determining the representation structure (e.g. nodes vs external state vector)
- Defining the granularity of knowledge components (vector dimensions vs number of nodes)
- Determining how much knowledge components are modified with the introduction of new knowledge
- How knowledge components relate to one another

Finally, since we are investigating personalised instructional sequencing for human students, we must also consider cognitive phenomena that affect our knowledge representation. In the next section I will introduce the role of forgetting and its role in personalised instructional sequencing.

2.3 Memory and forgetting

An important part of instructional sequencing is understanding the effects of time. More specifically, understanding and accounting for the forgetting that may occur over time. This is a simple concept with a rather complex and deep research history. This section

presents a very brief and simple overview of memory and forgetting in the context of learning. As this research is intended for a computer science audience, I do not delve too deep into the psychological and cognitive specifics of memory but rather try to provide an overview that is sufficient to understand the basics of memory and forgetting in the context of personalised instructional sequencing.

As we learn, there are certain bits of information we retain and some that we forget. An integral part of personalised instructional sequencing is approximating what information a student remembers and what information a student needs to revise to prevent forgetting.

The phenomenon of forgetting was first explored by Hermann Ebbinghaus in the late 1800s. Between 1880 and 1885, Ebbinghaus ran an experiment on himself to memorise a series of nonsense syllables (Ebbinghaus, 1885b). Through self-experimentation, Ebbinghaus uncovered the *forgetting curve*. The forgetting curve shows the decline of memory retention overtime. Specifically, it shows how information is lost over time if there is no attempt to retain it. There are many equations proposed to approximate the forgetting curve, but the simplest is the exponential curve. Whilst Ebbinghaus studied the nature of forgetting, his work was not focused on what instructional techniques should be used to counter the effect of forgetting.

One of the main methods of countering forgetting is *spaced repetition learning* (Landauer et al., 1978). Spaced repetition is a method of reviewing information repeatedly with increasing time intervals in order to minimise the effect of forgetting. There are several theories behind why this technique is effective. One of those theories is known as contextual fluctuation.

Context is all the stimulus that is present at the point in time that information is being reviewed. As time passes, context changes. The greater the time lapse, the greater the context change. For retrieval, it is believed that we rely on contextual cues. For example, if you have misplaced your keys, in order to remember where you have left them, you retrace your steps and try to remember what you were doing earlier. The activities you did earlier are the contextual cues. The more cues we have related to the particular piece of information we wish to recall, the easier it is to remember. Therefore, through

spaced repetition, information is encoded with increasing time intervals and thus varied contextual cues, making it easier to remember with the passage of time.

The study of forgetting is in fact the study of retention, as what is forgotten cannot be explicitly measured. Furthermore, what is remembered is only a reflection of what is remembered at a particular point in time. It is possible that at one point in time, the student cannot remember a definition of a word, but later, under different circumstances, he/she is able to successfully recall the meaning of the word. Rubin et al. (1995) suggests that ultimately it is not time that most directly influences forgetting, but it is how that time is filled. More specifically, interference or competing activity in period between two testing sessions is what determines retention. However, in an experimental setting, we must rely on a time as a proxy for interference.

Chapter 3

Instructional sequencing for vocabulary learning

In Chapter 2.1, I provide a historical account of instructional sequencing starting from the ancient Greeks to the modern era of AI and computers. In this chapter, I demonstrate how we can leverage ideas introduced in Chapter 2.1 and combine it with relatively modern methods such as reinforcement learning to develop an instructional sequencing system for vocabulary learning.

This chapter was adapted from work that was published and presented at Visually Grounded Interaction and Language workshop in the 31st Annual Conference on Neural Information Processing Systems (Zaidi et al., 2017).

3.1 Introduction

With the rise of machine learning and tasks such as automated teaching and assessment, there is an increased interest in understanding how machine learning models can be grounded in theories of learning. Additionally, with an abundance of learner data in archive and generation, we now have an avenue through which we can not only evaluate our theories of learning, but also explore whether these theories can be used to build better intelligent tutoring systems (ITS).

In this thesis I am particularly interested in the process of instructional sequencing for language acquisition and the role of machine learning in scaling its efficiency and reach. Language acquisition is a multidisciplinary field that overlaps with linguistics, psychology, neuroscience, philosophy, and more recently computer science. At the intersection of language acquisition and pedagogy lie theories of educational practices for language learners, including for example, an optimal curriculum for both L1 and L2 learners. L1 is a learner's first language, whereas L2 is the learner's second language.

A curriculum is a guide that helps teachers decide what content to present and the order in which it needs to be presented. The aim of a curriculum is to provide a highly structured series of concepts in order to maximise the rate of learning, develop a deeper understanding of the material and ensure long term retention. Instructional sequencing on the other hand, is the action or the act of optimising a curriculum to achieve the aforementioned objectives. In this thesis, a curriculum is seen to be an output of instructional sequencing. An optimal curriculum therefore is achieved through an optimal instructional sequencing methodology.

The idea of using a curriculum in an educational setting dates back centuries to the ancient Greeks. The first formal graded plan for group instruction, however, more likely dates back to the Lancasterian Monitorial instruction in the early 1800s (Saettler, 2004). Thereafter, it has been discussed and explored extensively. For example, Bruner (1960) introduces the idea of a "spiral curriculum", a curriculum in which complex information is first presented in a simplified manner and then revisited at a more difficult level later on. Vygotsky, from the perspective of language acquisition, introduces the idea of *scaffolding* within the curriculum in order to provide contextual support for more complex ideas using simplified language or visuals (Vygotsky, 1978). As we know from Chapter 2.1, the idea of teaching vocabulary with visuals was famously used by Comenius in 1658. His book, *Orbis Pictus*, was the first widely published children's textbook with pictures (Comenius, 1887). Although he did not formalise the concept of scaffolding, Comenius did recognise the importance of using visuals and other mediums to aid the learning process.

Even though it is not the primary focus of this thesis, it is worthwhile to mention

that theories of learning have also contributed to design decisions in training machine learning models. Elman (1993) draws parallels between the effectiveness of staged learning in humans, and in artificial neural models. More specifically, he describes “the importance of starting small” or going from simple to complex in a neural network paradigm. He coins the term “curriculum learning” to describe the process of going from simple to complex during the machine learning training process. This line of work was continued by Bengio et al. (2009), who illustrated how curriculum learning can facilitate the generalisation as well as the rate of convergence and training of deep learning networks. Hermann et al. (2017) also illustrate its benefits in the rate of learning for agents in a 3D simulation. Elman (1993) is evidence for why understanding how we learn is important for both education of humans but also for the training of machine learning algorithms.

In the classroom setting, Bruner (1961) argues that the role of the teacher is not to present information by rote learning but rather facilitate the learning process in order to teach students to become *active learners*: put simply, they are “learning to learn”. There are many factors that teachers need to consider when constructing a curriculum to achieve this goal, one of those factors is difficulty.

However, as we know from Pestalozzi’s teachings and later on in Bloom (1984), different learners have different abilities and therefore require a different order of instruction. In order to enable this, we have to consider a method of developing a personalised instructional sequencing system that incorporates Bruner (1961)’s proposition of difficulty, but can be developed over time to deploy, test, and evaluate other theories of learning such as Gagné et al. (1985)’s learning hierarchy. Presenting information from easy to difficult is a good choice for an instructional system but how do we know what is easy and what is difficult?

3.1.1 The notion of difficulty

One way of measuring difficulty of content is its position relative to the *zone of proximal development (ZPD)*, introduced by Vygotsky (1978). ZPD is a representation of what a learner is capable of achieving without help, with *some* help, and what concepts are beyond the learner’s current ability. However, the ZPD is an abstract concept that cannot directly

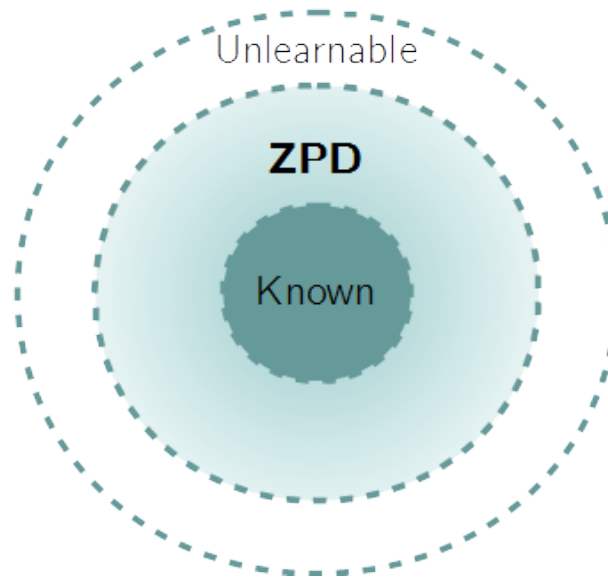


Figure 3.1: zone of proximal development (ZPD)
Source: Russell Moore and Ahmed Zaidi ALTA Annual Review 2017

be measured or determined. Therefore we have to find some method of approximating it either heuristically or algorithmically.

A common method of identifying difficulty of content is by conducting laborious and resource intensive studies which involve experts carrying out focus groups and analysis to decide where a particular question or topic sits in the curriculum. Once that question has been placed into the curriculum, it is tested and revised periodically. Revisions are a result of a certain percentage of students getting the question right or wrong. For example, if only 5 to 10 percent of the students answer the question correctly, it is likely to be too difficult for that target group of students. This method is highly inefficient as it results in questions being inaccessible and wasted for a large proportion of the students. In a personal instructional sequencing system, the system should have an efficient method of measuring difficulty that evolves with the learner's ability and ensures a difficult question is only difficult with respect to the ability of that learner. Furthermore, the degree of difficulty is such that with some support the learner can grasp that concept. This is in line with Bruner et al. (1966), stating that content should be created in such a way as to create uncertainty for the student which in turn elicits curiosity within the learner. One of the ways of creating uncertainty within the learner is ensuring optimal difficulty of

content.

There are several theories in addition to ZPD that describe how to think about the difficulty of content with respect to the learner. Another concept introduced by Krashen (1989) is the input hypothesis or the $i+1$ theory. This hypothesis states that in order for students to progress their linguistic ability, they must learn content slightly more advanced than their current level. Krashen and Terrell (1983) also introduces another hypothesis, the *natural order hypothesis*, that states language should be taught in a pre-specified “natural” order that remains the same between different learners. This sounds somewhat contradictory to the overall purpose of this chapter and thesis and has also been challenged by (Murakami and Alexopoulou, 2016). The question that arises from the natural order hypothesis is as follows: is personalised instructional sequencing about pace and rate of progression through the content or is it about the overall order of the content as well? It is worth noting that the natural order hypothesis is specifically for language acquisition. We will explore this in further detail in the discussion section of this chapter.

3.1.2 Task description and contributions

I have described the motivation for a personalised instructional sequencing system in Chapter 1. This is largely supported by the findings of Bloom (1984) and the *Two Sigma Problem* stating that individualised instruction yields the best learning performance. Creating a personalised instructional sequencing system requires some notion of content difficulty which is typically a resource intensive task. Therefore, to address these limitations, I used a reinforcement learning framework to learn difficulty of content for each student. This would allow the difficulty of the question to be implicitly measured with respect to the student’s current knowledge state. The task I use to demonstrate this methodology is vocabulary learning. As mentioned in Chapter 2.2, knowledge state is composed of many knowledge components. In this task, I have assumed each knowledge component is a single vocabulary word. Inspired by Comenius and his book, *Orbis Pictus*, I also use images as the method with which to teach vocabulary words. The objective for the student is to enter the word that correctly corresponds to the image presented. The

goal of the reinforcement learning framework is to learn the optimal sequence of words that the student should be presented in order to ensure best learning outcomes. The framework must infer the correct order of content automatically. In the next section I will introduce reinforcement learning and describe how I define optimal in the context of the task.

The main contributions of the this chapter are as follows:

1. I present a novel Q-Learning framework for the task of instructional sequencing that models student ability through variable reward. Previous uses of reinforcement learning for instructional sequencing relied on a dynamic state representation which is then used to determine the optimal action using an existing policy. In my approach, I assume static state representations but adapt the reward according to the student proficiency.
2. I introduce a novel vocabulary learning system where I incorporate the use of language models and image recognition systems to enable the learning and assessment process.

3.2 Reinforcement learning

The reinforcement learning (RL) framework is the backbone of the personalised instructional sequencing framework. However, let's first describe the basics of reinforcement learning (RL) so that it is clear how it is being applied to personalised instructional sequencing. RL is a subset of machine learning that aims to train an agent to take the actions within an environment that maximise its long-term cumulative reward (Sutton and Barto, 2018). Typically, RL is represented in an Markov decision making framework which is defined as follows:

S = state/observations in the environment

A = the set of actions an agent can choose from within the environment

$R(s, a)$ = reward from taking an action in a given state

$T(s'|s, a)$ = transition probability of going from s to s' given an action

Our objective is to learn an optimal policy π . A policy is a guide or map that provides the agent with the action it should take given the state it is currently in such that it maximises the long-term cumulative reward. Broadly speaking, there are three different approaches to RL: policy search approach, value-function based approach and model-based approach (see Figure 3.2). Policy search and value-function approaches rely on trial-and-error to converge on the optimal policy. That is, they use an approach which learns what reward would be obtained by trying out different actions in a particular state. This approach is necessarily conducted over several iterations. The algorithm typically acts greedily or in other words, takes the action with the highest immediate reward. However, this can quickly result in the agent getting stuck in a local optima if it only knows the reward associated with one of the possible actions in the given state. This phenomenon is known as the explore-exploit problem. That is, the model must decide whether to take the action that has the current maximum reward, or take a new action that it has never seen before. Usually algorithms behave probabilistically, where they will for example, 80% of the time take the action with the highest reward and 20% of the time take a new action.

On the other hand, a model based approach relies on having a model of the environment that mimics the behaviour of the real environment. In the model based approach, the optimal policy can be determined prior to taking an action given the modelled environment. Each of these approaches have their strengths and weaknesses. Let's consider each in term:

3.2.1 Policy search approach

As the objective of an RL algorithm is to find an optimal policy, the most direct way of reaching that goal is to use the policy search approach. However, it turns out that trying to search for the optimal policy directly is quite difficult. This is because in certain

cases, we do not have access to the data that help us determine whether the action we took was optimal given the state. The original method of using the policy search method was introduced by Howard (1960a). The method was developed for MDPs. However, in certain tasks, we cannot directly observe the states of an MDP and therefore must rely on methods for partially observable Markov decision processes (POMDP) (Astrom, 1965). The knowledge state of a student would be an example of a state that we cannot directly observe.

3.2.2 Value-function approach

An alternative approach we can take is the value-function based approach. An important differentiation to note is between the concepts of reward and value. A reward is what an agent receives from the environment when an action is taken in a given state. A reward is immediate. Value, on the other hand, is more long term, and provides a signal of the overall rewards that may accumulate as a result of being in a particular state. It is only by knowing the approximate value, not immediate reward, of state-actions pairs that we can map to an optimal policy π .

3.2.3 Model based approach

The last approach is the model based approach. This is where we have a model that mimics the behaviours of the environment. For example, given a state and action, the model can predict the next state and reward. This will allow us to make inferences about what actions to take in the future without having to actually observe them. In Figure 3.2 T refers to the transition function which predicts the next state given the current state and action. R is the predicted reward returned from the environment given a state-action pair. In reality, having an accurate model of the environment a priori is not always possible. In those scenarios we must rely on model free approaches such as policy search and value-function based approaches.

In RL, the most important aspect is the algorithm's efficiency to measure value.

Therefore, most of the research in RL has been focused on value function based approaches. It is value, not reward, that should be maximised when selecting the best state (Sutton and Barto, 2018).

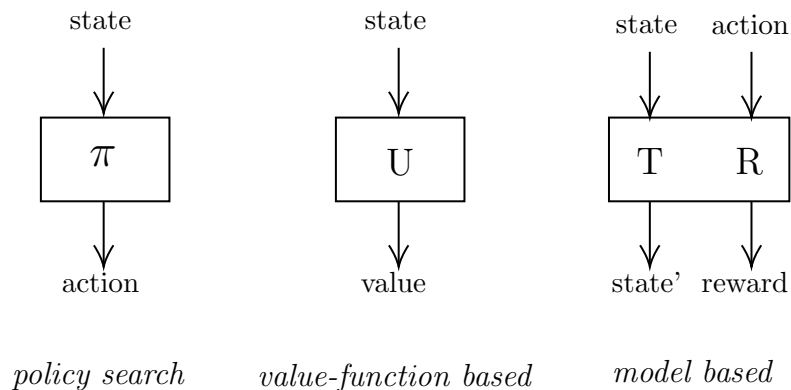


Figure 3.2: A figure showing the three different approaches to RL, where π stands for policy, U stands for utility, T stands for transition function and R stands for reward function.

Let's now consider how RL relates to the instructional sequencing task. There are many features of RL that are shared with instructional sequencing. For example, when learning a language, we must not only arrange our revision schedule to remember a particular word tomorrow, we must consider how we can retain the meaning of this word in the long-term. This is similar to the reward vs value distinction made earlier.

When tutoring a student, the teacher relies on a noisy signal of the student's cognitive and knowledge state to make decisions on the next best action. These signals are usually captured through assessment events (AEs) such as a test, a quiz, or conversations as described by Koedinger et al. (2010). Similarly, in a POMDP RL environment, since we cannot directly observe the state, we must rely on noisy observations of the state to make assumptions of the optimal actions to take.

The use of RL in personalised instructional sequencing is not a recent innovation. Smallwood (1962) was one of the first to use RL for instructional sequencing. In fact, according to Doroudi et al. (2019), instructional sequencing was one of the first applications of RL in general. In Smallwood (1962), the author describes the parallel nature between MDPs and instructional sequencing. To learn more about this, refer to Chapter 2.1.4.

Other examples of RL in pedagogy include Atkinson (1972b) who demonstrates its

application in optimising the teaching of German vocabulary. Beck et al. (2000) uses RL to teach students arithmetic, aiming to minimise the time taken to answer questions. Iglesias et al. (2009, 2003) teach students database design using Q-learning, a specific type of RL framework. Both Beck et al. (2000) and Iglesias et al. (2009, 2003) evaluate results on simulated students. Martin and Arroyo (2004) use RL for teaching maths while Tetreault and Litman (2006) use it for teaching physics. The approach proposed by Rafferty et al. (2016) to use POMDPs for faster teaching resembles the approach and task we are addressing in this chapter. However, in contrast to Rafferty et al. (2016), where the reward function and values are fixed once the policy converges, in this work, the reward function is constantly updating in line with student progress. Furthermore, where the former work learns a student model or state, in my work, I assume discrete student states that are predetermined.

As far as I know, no previous work has been done in exploring the use of RL for vocabulary acquisition with images where the student progress is modelled in the reward function. However, Whitehill and Movellan (2017) present a framework that is tested on the word-image matching task. Whitehill and Movellan (2017) extend the work of Rafferty et al. (2016) by using a student model that allows teaching actions that only partially eliminate the student’s beliefs about the words’ meaning. The authors also incorporate a deeper search through possible learning trajectories for the student. For a more comprehensive literature review on RL in instructional sequencing, please refer to Doroudi et al. (2019).

In order to automate the process of instructional sequencing for visual vocabulary acquisition, I must first identify the key components of our RL system. The agent in this task is the automated tutor that must learn what information to present to the student. The environment is the student with whom the agent is interacting.

3.3 CEFR and Cambridge Learner’s Dictionary

I assume that the student is a learner of English who has reached a certain level on the *Common European Framework of Reference* (CEFR) scale. CEFR is an international standard for describing language ability, using a six point scale, from A1 for beginners, up to C2 for those who have mastered language (See Figure 3.3).

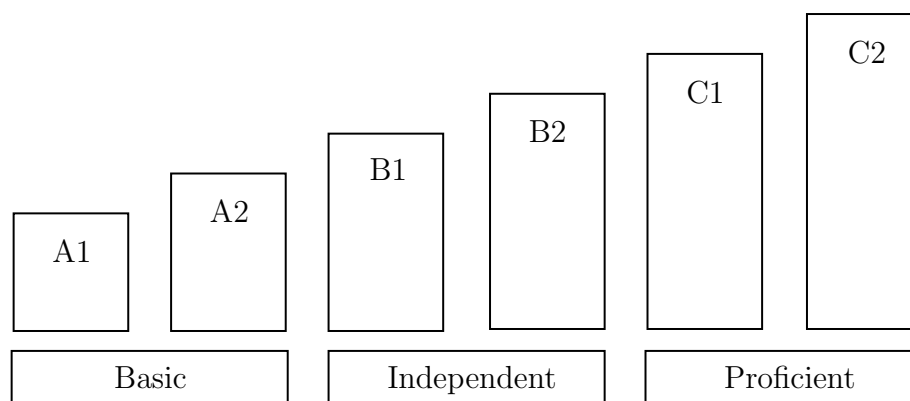


Figure 3.3: Common European Framework of Reference

In order to obtain the CEFR level for each word, I used the Cambridge Learner’s Dictionary, which is a dictionary that contains words with their corresponding CEFR level. I cross referenced the list of words in the CEFR dictionary with the list of words included as labels in our image recognition model VGG-16 (discussed further in 3.5). This resulted in 243 words with their corresponding CEFR levels that are also recognisable by an image recognition model. To expand on this data set, additional synonyms were generated automatically for each of the target words by outputting the top 10 nearest words to the target word in a pre-trained skip-gram word2vec model Mikolov et al. (2013a). These additional words were cross referenced in the Cambridge Learner’s Dictionary to obtain their relevant CEFR levels. This enabled students to enter the target word or a relevant synonym with presented with an image. One of the limitations of this approach was accounting for words with multiple senses e.g. bank. “Bank” for money is A1 where as the river “bank” is B1. In the Cambridge Learner’s Dictionary the word tense with the lower CEFR appears first and that is the word tense I used. One of the assumptions I make by utilising the word2vec model to increase possible acceptable answers is that the

student will only enter relevant words. This is an important assumption as the 10 nearest words to the target word may include opposite or semantically different words e.g. cat and dog.

3.4 Curriculum Q-Learning

Thus far we have discussed the various methods and approaches used for instructional sequencing. I will now describe RL algorithm used to identify optimal order of instruction for a given student. The RL algorithm used by my proposed system is a modified Q-Learning algorithm. Q-Learning can be defined as follows:

$$Q^{new}(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[(r + \gamma \max_a Q^\pi(s_{t+1}, a)) - Q(s_t, a_t)] \quad (3.1)$$

where $Q^{new}(s_t, a_t)$ is the new Q-value of state s_t and action a_t tuple at time t and $Q(s_t, a_t)$ is the current Q-value of state s_t and action a_t tuple. The $\gamma \max_a Q^\pi(s_{t+1}, a)$ is the maximum expected reward from the future state s_{t+1} and all of its possible actions. The α is the learning rate and γ is the discount factor. γ models the fact that future rewards are less valuable than immediate rewards at a given time t .

A policy π maps states s to actions a . The aim of the Q-Learning algorithm is to find an optimal policy π such that it maximises the long-term cumulative reward. The policy achieves this by acting greedily and taking the action that presents the maximum Q-value given the state such that $\max_a Q^\pi(s_t, a)$.

In action selection, there is a trade-off between *exploiting* what you have learnt so far and *exploring* other state-action tuples. In this task I model that using ϵ -greedy. This means the policy will, for the most part, select the actions that provide the highest estimated future reward given the state. However, with a probability of $1 - \epsilon$, an action will be selected randomly and independently from a uniform distribution. Action selection is drawn from a Q-Table which is a table that stores all state-action Q-values.

In this task, a policy can be viewed as a personalised instructional sequence as it decides what should be shown and in what order. In order to learn the optimal sequence

of vocabulary words I leverage the structure provided to us by the CEFR framework. The optimal sequence can be defined by the sequence of words that result in the shortest time to learn all the words. This reinforcement learning based instructional sequencing algorithm behaves similar to the drill-and-practice approach introduced by (Atkinson and Raugh, 1975).

In this framework I have 6 levels and within each of those 6 levels I have a number of words. Therefore, I incorporate two models, the *CEFR level model* and *word level model*.

CEFR	Back	Remain	Next
A1	0	1	0
A2	0	1	0
B1	0	1	0
B2	0	1	0
C1	0	1	0
C2	0	1	0

Figure 3.4: Common European Framework of Reference Q-Table

Status	Remain	Toggle
Active	0.1	0
Inactive	0	0

Figure 3.5: Word Memory State

In the CEFR model (see Figure 3.4) there are 6 states, where each state is one of the CEFR levels. The actions are whether the student should progress to the next level (referenced as Next in Figure 3.4), remain in the current level (referenced as Remain in Figure 3.4), or go back a level (referenced as Back in Figure 3.4). Since the actions only enable going forward, remaining or going back a level, it is not possible for a student in state A1 to jump to state C2. The word level model (see Figure 3.5) has two states: Active (show the word), Inactive (hide the word). The two actions are, Remain in the

current state or Toggle state. This architecture ensures that there is also an estimated long-term reward (Q-value) associated with showing a student a particular word.

Modelling reward is often viewed as a challenging task in RL as it requires careful consideration on implications the reward will have on agent behaviour. For this application, a student is rewarded negatively (-1) for getting a question correct and positively (+1) for getting it incorrect. The motivation behind using these rewards is grounded in the idea of diminishing returns. The RL model acts greedily and takes the action with the maximum long-term reward, so if I review a concept I understand, then I am not gaining knowledge by reviewing it again. Thus, its value should be reduced. Alternatively, if I get a question wrong, the benefit of reviewing that word is higher, and thus I should increase the associated Q-value.

3.5 Student response

To evaluate the students' understanding and learn a policy, I present a word in the form of an image. The objective for the students is to describe the image, and based on their response, the Q-Table and thus the policy is updated. A valid response is defined by the target word associated with the image or a near synonym of that target word. One of the key requirements for the personalised instructional sequencing system is to ensure that the system is able to handle the varied response from the students and to make it simple and efficient to use for the instructor.

The instructor can simply insert relevant images for the target group in a pre-specified folder. Using a pre-trained image recognition model VGG-16 (Simonyan and Zisserman, 2014) each image is labelled with list of words associated with the image.

3.6 Visual vocabulary instructional system

The personalised instructional sequencing system was built using Python based micro web framework Flask. Using the framework an API was built that connected to a front-end user

interface (see Figure 3.6) and a SQL database that saved the progress for each student. In Figure 3.6, the UI also provides visibility on CEFR level the student is currently performing at as well as the associated Q-values. There is also a function to probe into the Q-value statistics associated with a particular word. The motivation behind presenting the Q-table to the student and teacher was to encourage model transparency and allow the teacher to understand why a student is at a particular CEFR level and what words in particular are holding back the student’s progress. Furthermore, even if the student progresses to the next CEFR level, the systems allows teachers to probe words in the CEFR level below to identify the knowledge components that the student has historically struggled with and reset those words Q-table’s. This automatically reintroduced those words into the personalised instructional sequence.

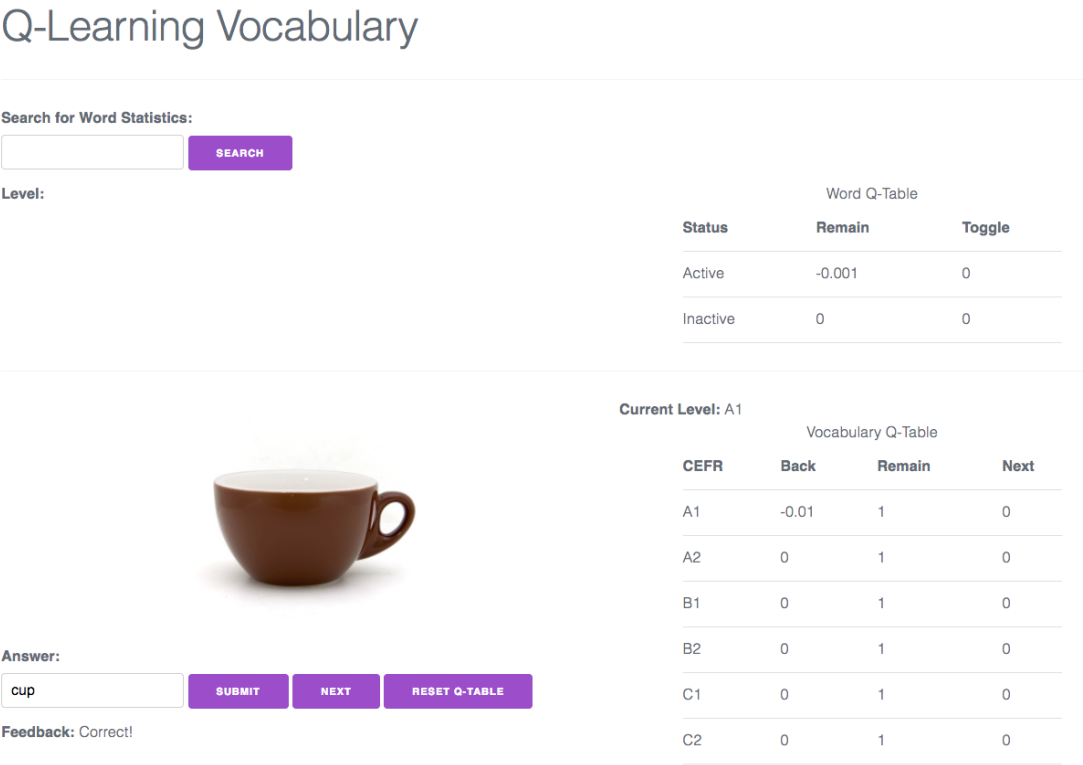


Figure 3.6: Personalised Instructional Sequencing System User Interface

As mentioned previously, when a student answers the word correctly, the Q-value of that word drops in value. However, the amount the value drops is incremental. The decision to make value updates incremental ensures that words are reviewed and practised several time before they are considered “mastered” and move into an inactive state.

3.7 Experiments

For the CEFR level model, I use a learning rate α of 0.1, a discount rate γ of 0.9 and an ϵ value of 0.95. The word level model uses an α of 0.1, a γ of 0.9 and an ϵ value of 1. The learning rate α is what ensures that the Q-value is updated incrementally. The purpose of the discount rate γ , besides that it provides mathematical convenience for the algorithm, is to determine how important future rewards are. The higher the value of γ the more important future rewards are. The ϵ value determines how often the algorithm should act greedily versus how often it should act randomly. The higher the ϵ value the more greedily the algorithm will behave. The reason why the word level model maintains an ϵ value of 1 is because it does not make sense for the agent to randomly toggle the word state inactive. The α and γ values were chosen in accordance with other work in the Q-Learning space that employed similar values (Littman, 1994, Russell and Zimdars, 2003, Bianchi et al., 2004, Matignon et al., 2007).

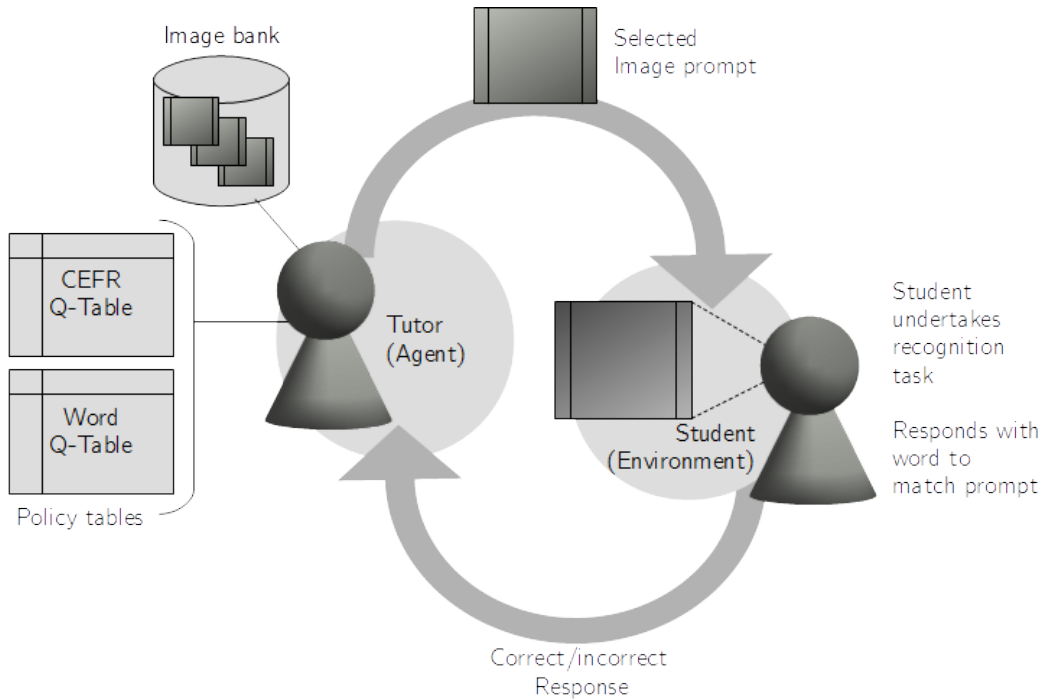


Figure 3.7: Visual Vocabulary Q-Learning Personalised Instructional Sequencing System

Let's consider an example of how Q-Learning algorithm works in relation to our vocabulary learning system.

1. **Initialise Q-Table.** At initialisation, I place increased value on the “remain” state in order to ensure that until the student has demonstrated that he/she understands the vocabulary words at this level, progressing to the next level is unlikely (not impossible due to the ϵ greedy value).
2. **Choose Action.** The action is selected based on which action has the highest Q-value in the given state: $\max_a Q^\pi(s_t, a)$ where s_t is A1. In this particular circumstance, “remain” action has the highest Q-value of 1. At initialisation, every student starts at the A1 state.
3. **Perform Action.** I selected on an epsilon value of 0.95. This means that 95% of the time the action with the highest Q-value will be chosen. 5% of the time a random action will be chosen. Assuming the action with the highest Q-value is chosen, an image corresponding to an A1 level vocabulary word is presented. The student then submits a response.
4. **Measure Reward.** If the student’s response is correct, a reward of -1 will be given.
5. **Update Q-Table.** In order to update the Q-Table, let’s refer to the Q-Learning algorithm referenced in Equation 3.1. In order to update the Q-value of the state action tuple just selected i.e. $Q^\pi(A1, remain)$, I must first calculate the Q-target value which is $r + \gamma \max_a Q^\pi(s_{t+1}, a)$. I replace the variables: $r = -1$, $\gamma = 0.9$, $\max_a Q^\pi(s_{t+1}, a)$ or the expected future reward given the new state and all possible actions of this state = 1. I arrived at the value of expected future reward by knowing that the next state will be “A1” as the action “remain” has the highest Q-value (value of 1). Therefore, the Q-target value is -0.1 from which I subtract Q-predicted or $Q(s_t, a_t)$ which is 1 resulting in -1.1 . I then multiply this value with the learning rate ($\alpha = 0.1$) and reach a final update value of -0.11 . Therefore, the new Q-value for $Q(A1, remain)$ is $1 + (-0.11) = 0.89$. Since I got the answer correct, the value of staying in A1 was reduced. Had the student submitted the wrong answer and

the system returned a +1 reward, the new Q-value for $Q(A1, remain)$ would be $1 + 0.09 = 1.09$.

3.7.1 Simulated Students

Reinforcement learning (RL) systems frequently suffer from the ‘cold start’ problem as it takes a long time to accumulate enough data to begin optimising a policy in a meaningful way. The number of episodes needed to initialise the model depends on the problem and algorithm being used. However, hundreds of thousands of iterations through the data is fairly typical. Therefore it is impractical to cold-start an RL system that is aimed at interacting meaningfully with human subjects. With such an approach we might expect the system to take months to converge on a useful policy using observations returned at human speed, and in the meantime the human guinea-pigs would be subjected to all manner of random exploratory behaviour and lose interest.

In this context, we are fortunate to have an existing structure and hierarchy in the data that enables less exploratory behaviour. Namely, the CEFR levels provide a predetermined order of instruction. Therefore, the state search space for the model has been reduced from the size of the total vocabulary to the number of the CEFR levels. Thereafter, the agent will be required to guide the students through the vocabulary words within the CEFR level. The agent is required to infer the difficulty of each word within the CEFR level not between them. This reduces the complexity of the task. Had I decided not to leverage the CEFR structure, learning an optimal policy would prove quite challenging and require many more iterations as each word would be considered an equally likely state. Therefore, one can argue we have taken a hybrid model-free and model-based approach. The model-based approach referring to the fact that we have leveraged CEFR levels as a signal for the underlying structure of vocabulary.

However, we still require simulations to observe how different students would interact with the agent. By pre-training with simulated students we can provide future human students with an option to self classify (beginner, intermediate, advanced). Based on their self classification, a pre-trained policy can be used. If that policy turns out to be

ineffective, I can compare their reward functions to the reward functions of the simulated students to determine their correct classification.

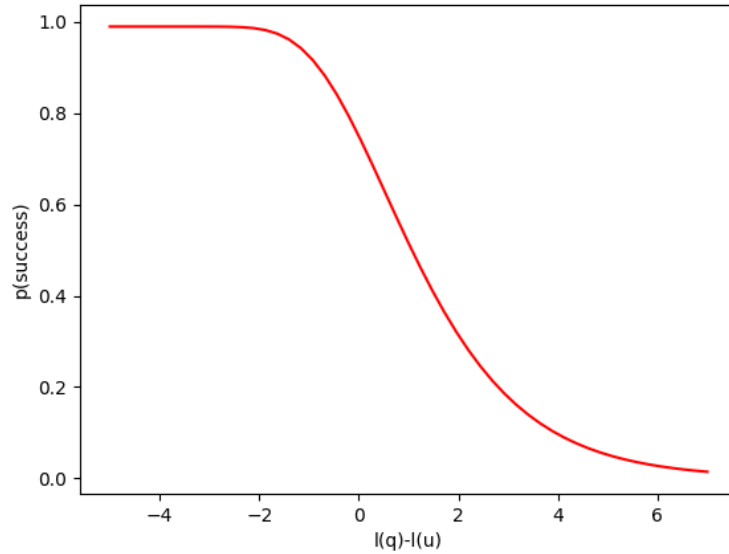


Figure 3.8: Gompertz Curve which estimates the probability that a student will correctly answer a question $l(q)$ given the level of the student $l(u)$.

To evaluate the performance of our system, I simulated three classes of students at varying levels of proficiency: *beginner*, *intermediate* and *advanced*. In this case, I modelled the student’s probability of getting a question correct as a negated Gompertz (Winsor, 1932) distribution:

$$P(\text{success} \mid u, q) = 1 - \exp(-b \exp(-c(l(q) - l(u)))) \quad (3.2)$$

where $l(u)$ denotes the level of user u calibrated to a scale of $[0, 6]$. Each integer in the scale represents a corresponding CEFR level from A1 to C2 (*e.g.* $0 \rightarrow A1$, $1 \rightarrow A2$, *etc.*). $l(q)$ represents the level of an item q (i.e. a word corresponding to the image) calibrated to the same scale. The parameter b determines the probability of success when student and item level match. This is set to $\ln(4)$ to model a ‘typical’ pass rate of 75%. The parameter c is the growth rate which was set to 1. The calibrated curve is shown in Figure 3.8. The curve is flatter at the lower end as students may be expected to be comfortable with most of the material at lower CEFR levels than their own, whereas at higher levels, their ability

is more uncertain. I ran simulations where I sampled 100 students from each student class (beginner, intermediate, and advanced). Each of these students had 100 interactions with the agent tutor. An interaction can be defined as when a student responds to a question. For the purposes of this simulation, I assumed that a beginner student was at a level between A1(0) and A2(1), an intermediate student was at level between B1(2) and B2(3) and an advanced student was at level between C1(4) and C2(5). Figure 3.9 to Figure 3.11 illustrate the behaviour of the Q-Learning policy for students at different levels of proficiency. This is compared to a baseline policy.

Figure 3.12 shows the average cumulative reward earned by students at varying levels of proficiency.

3.7.1.1 Baseline Tutoring Policy

In order to evaluate the performance of the Q-learning tutoring policy I compared it to a baseline policy. The baseline policy is inspired by mastery learning where the student can only progress to the next level if the student has consecutively answered 10 questions correctly at the current level.

3.7.2 Results

Student Level	Q-Learning Iterations	Baseline Iterations
Beginner	31	52
Intermediate	48	82
Advanced	131	94

Table 3.1: Number of iterations taken by Q-Learning Model and Baseline Model to converge at student level

In Table 3.1 we can see that the Q-learning model converges to the actual level of the student far fewer iterations than the baseline model, specifically for beginner and intermediate students. For advanced students, the baseline model outperforms the Q-learning model. The results in Figures 3.9 to 3.11 show on average how the agent tutor responds to the three classes of simulated students at various proficiency levels. The

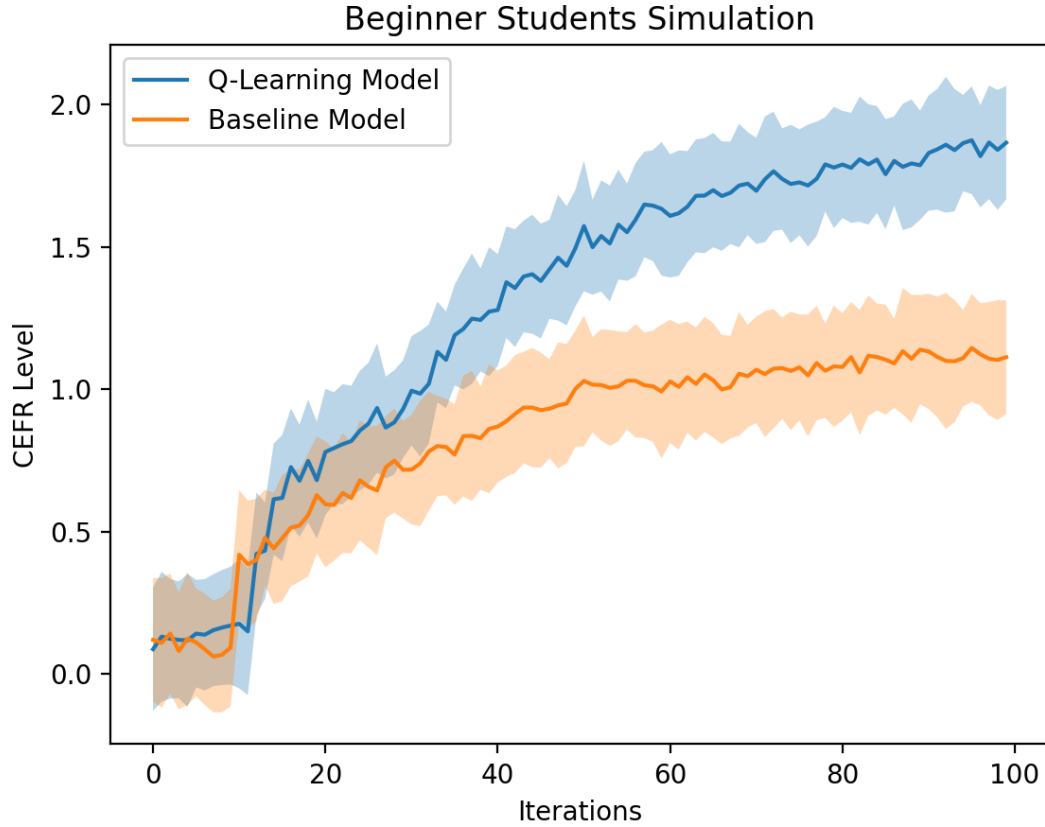


Figure 3.9: CEFR levels determined by the agent for beginner students over 100 interactions.

beginner students fluctuates between A1 and A2 which is as expected. In comparison the baseline model remains in A1. The agent tutor presents intermediate students with content that lies between B1 and B2. The baseline model remains on the other hand does not push the student beyond B1. For advanced students, the agent tutor presents vocabulary words that are at the B2/C1 level. In this case, the baseline model pushes the students by presenting vocabulary words at a C2 level. We can also see that the agent tutor sometimes behaves randomly due to the ϵ value but the students eventually converges at their actual level and ZPD or the edge of the curriculum. Figure 3.12 illustrates how the cumulative reward of students varies at different proficiency levels. The curve experiences a downward slope as the students reach their current level of vocabulary; at which point reviewing material at their level of proficiency results in diminishing returns. If these students were human students, ideally we would observe a trend similar to that of advanced students in

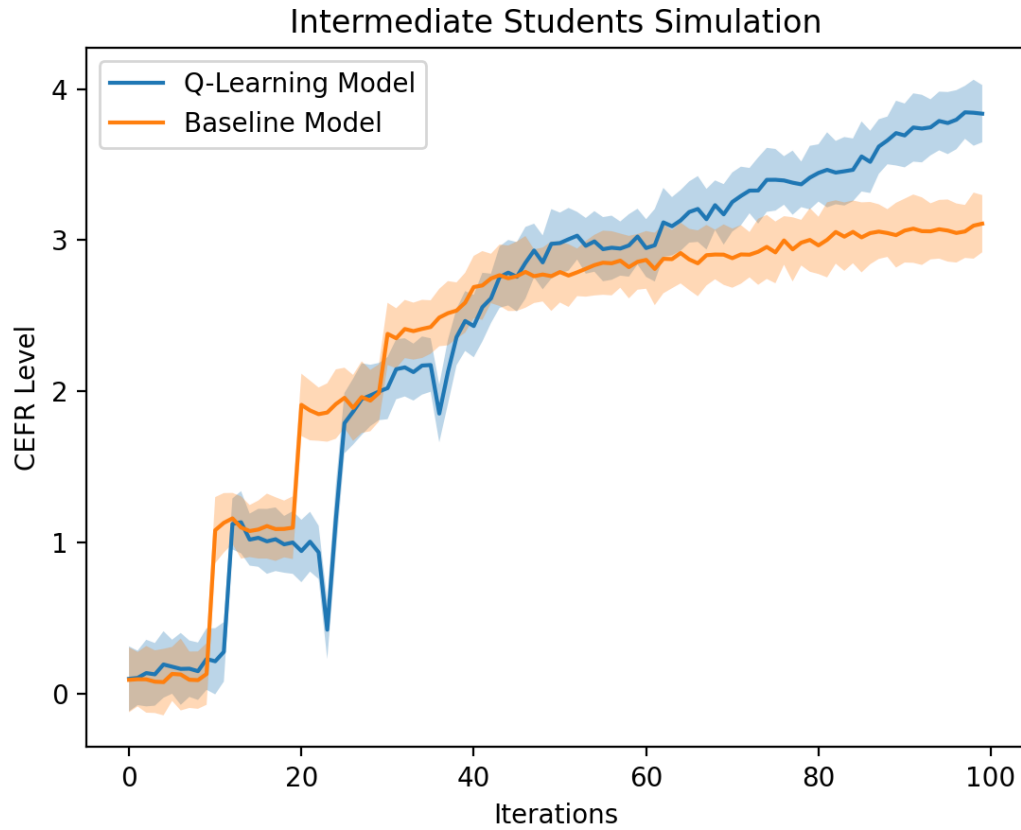


Figure 3.10: CEFR levels determined by the agent for intermediate students over 100 interactions.

Figure 3.12. For human intermediate and beginner students we might hope to observe an increase in their cumulative reward especially as they approach their ZPD. In theory, these students will reap the most value from reviewing the information they are unfamiliar with. Instead, in the case of simulated students as seen in Figure 3.12 you notice a decline in their cumulative reward due to the fact that they already know all the words at their predefined level. This suggests that while the system is effective at identifying the optimal level for a particular simulated student, it is important to test this on human students with whom we will observe learning gains.

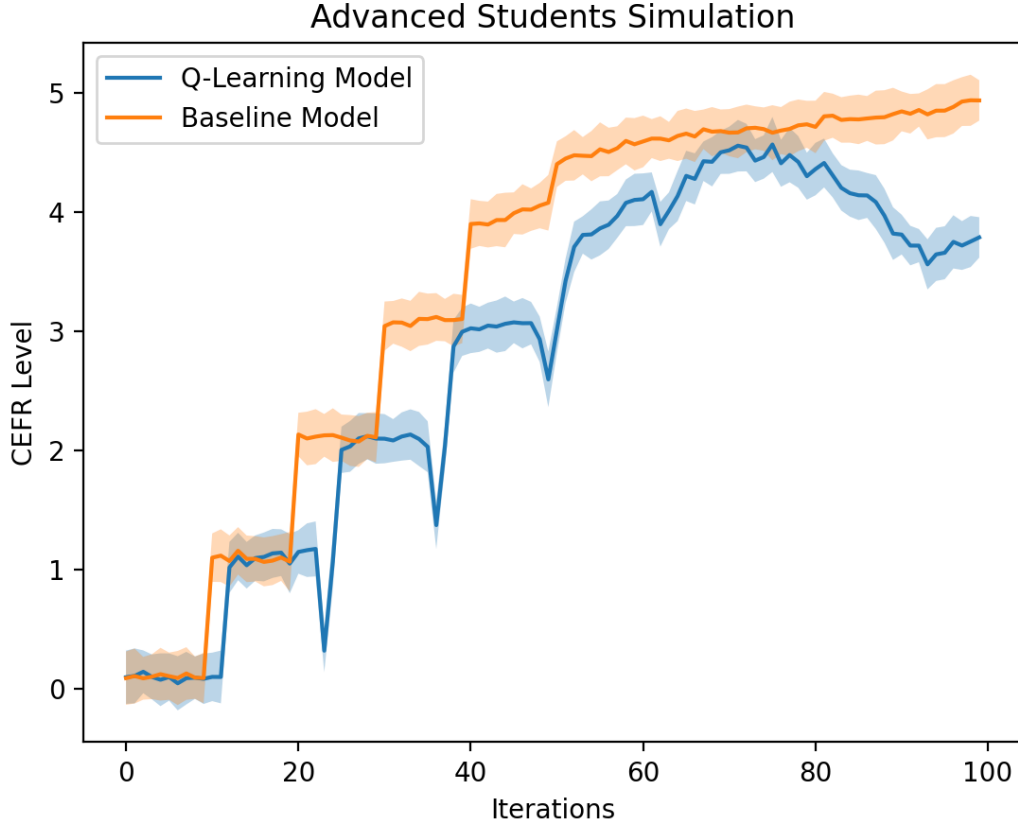


Figure 3.11: CEFR levels determined by the agent for advanced students over 100 interactions.

3.8 Discussion

I have shown through the use of simulations, that we can effectively model the vocabulary learning task as an RL system. Figures 3.9 to 3.11 and Figure 3.12 show clear indications of varying agent behaviour for students at different levels of lexical proficiency. When compared to the baseline approach, the Q-learning model is able to adapt to the needs of the beginner and intermediate students more effectively. This is demonstrated by the fact that students are being pushed within the current level. We can also observe that the Q-learning model for beginner and intermediate students is able to reach the actual level of the student in less iterations than the baseline approach. Furthermore, the Q-learning model has the ability to move down a level automatically depending on the performance of the student.

Beyond that, I have set up a framework that can also be used in the future to extrapolate

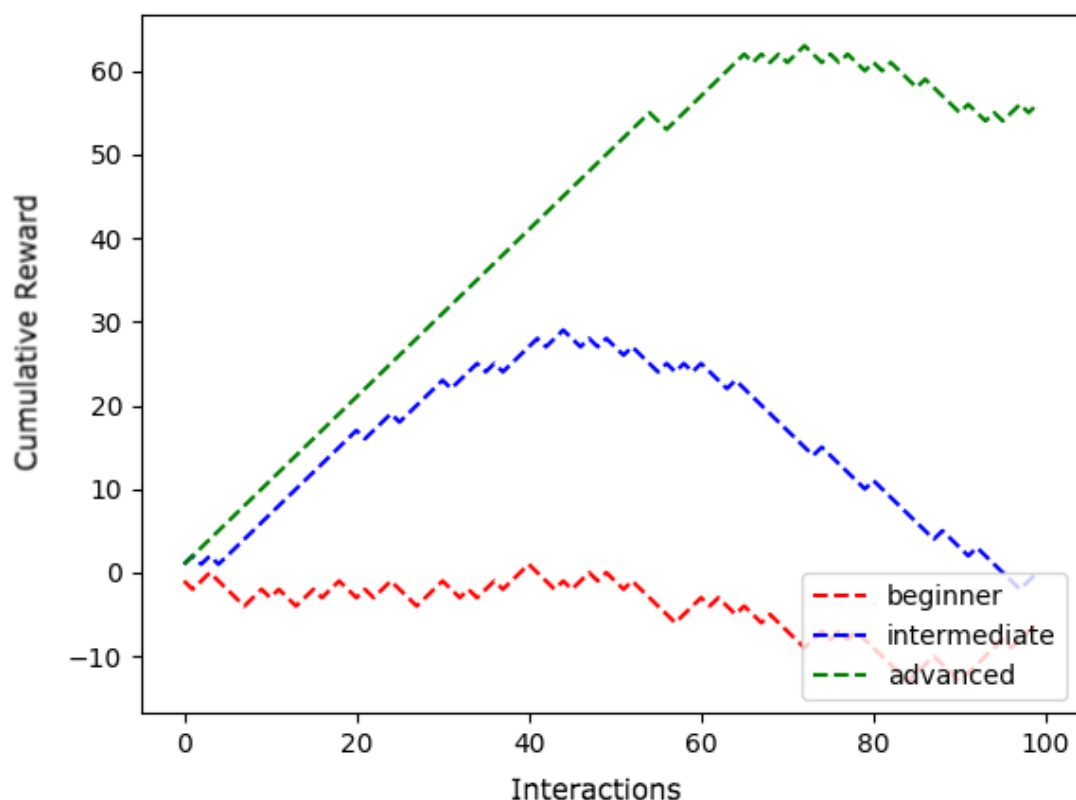


Figure 3.12: Cumulative reward earned by students of varying levels of proficiency from the agent over 100 interactions.

the difficulty of new material. This system will serve as a test bed that will yield metrics to determine where the content fits in the curriculum. Although this is foundational work, it lays the building blocks for future pedagogically inspired RL architectures.

Through this work, I have also shown that there are many similarities between the principles of RL and theories of language acquisition. Specifically, parallels can be drawn between the concept of ϵ -greedy and Krashen's Input Hypothesis or the $i+1$. The Input Hypothesis states that students learn by comprehending language that is slightly above their current language level. The interactions between the agent and the environment in RL is analogous to the social interaction approach to language acquisition, specifically the equal importance of input and output. Furthermore, our use of a model-free RL approach with limited structure provided by CEFR levels, reflects the nature of learning. Mainly, it outlines that fact that we are still a long ways away from fully understanding

how humans learn and thus require a model-free approach. However, over the years, we have found hints and evidence about certain aspects of learning, e.g. learning should follow the natural order of development (simple to complex) and therefore, finding ways to incorporate those insights into the model-free model will improve efficiency and likely performance. I illustrated this by dividing words by their CEFR level.

However, there is scope for substantial extensions in this space. Deploying the system on-line in order to collect user data will allow us to validate and improve our existing models. Incorporating memory and spaced repetition learning (Settles and Meeder, 2016), a phenomenon initially documented by Ebbinghaus (1885) and discussed in Chapter 2 and later in Chapter 5, in order to optimise the policy and emulate cognitive processes is also an important extension that may have a great impact on the learning output. Furthermore, extending the framework to incorporate other tasks is also important as the use of just images is quite limited. As the vocabulary becomes more complex, it becomes increasingly difficult to rely on images as a medium for learning because many words are difficult to represent visually. Furthermore, the use of image recognition systems also pose potential limitations due to the way they are trained. Image recognition systems are trained with a classification objective which means that any vocabulary words the instructor desires to be included as part of the curriculum must have been part of the training set for the image recognition system. Although I circumvent this limitation through the use of word embeddings and semantically similar words, it is still a considerable limitation to the existing system. Experimenting with other task types would also be beneficial in terms of evaluating the robustness of the teaching policies learnt using the RL system. Finally developing improved models of representing student knowledge states will also be critical in improving the effectiveness of RL and personalised instructional sequencing systems. I will discuss the development of student representations in Chapter 4

Finally, all of these models can also be applied to machine learning algorithms instead of students. As discussed previously, Hermann et al. (2017) indicated the need for a curriculum in order to effectively train an agent in the simulated environment. Guiding an RL agent with a policy that is inspired by theories of learning may result in improved

learning rates.

Chapter 4

Student modelling

In the previous chapter, I presented an example of how Q-learning can be used to develop a personalised instructional sequencing model for visual vocabulary learning. I proposed the use of a Gompertz curve to simulate students for the purposes of evaluating our system. However, the Q-table model as a representation of the student’s knowledge, while highly interpretable, is also quite limited. In this chapter, I explore alternative methods of developing improved student representations that will not only provide us with more information about the student’s ability but also the potential to improve instructional sequence inference.

This chapter was adapted from work published and presented in the 12th International Conference on Educational Data Mining (Zaidi et al., 2019).

4.1 Introduction

I define personalised instructional sequencing as a computational procedure for the automatic selection and presentation of teaching materials which are deemed most suitable for the user at a given point in time. In this framework, the platform user – a STUDENT – is guided through online courseware – a CURRICULUM – in an optimal and personalised fashion. In order to select items (TASKS) for students appropriately it is necessary to relate accurate machine-readable representations of each individual task to machine-readable

representations of each student¹. Such representations can be used to predict future performance on parts of the curriculum that a student is yet to reach (as in Minaei-Bidgoli et al. (2003), Khajah et al. (2014), Zaidi et al. (2017)). These predictions can in turn be used to select the set of appropriate next items for this individual – those which are not too easy and not too hard (as in Adomavicius and Tuzhilin (2005), Fudholi and Suominen (2018)).

In general the personalised instructional sequencing approach has been shown to lead to improved learning outcomes for student users of educational platforms (Lindsey et al., 2014a, Najar et al., 2016, Rosen et al., 2018). In Chapter 2.1.4 I provided detailed overview of both instructional sequencing and knowledge representations from a historical perspective.

In summary, rich knowledge representations are critical to developing better instructional sequencing models. As I discussed in Chapter 3, one of the challenges of the RL framework is what is often referred to as the multi-arm bandit problem or the explore versus exploit conundrum. What this implies is that unless the student has interacted with all parts of the domain or curriculum, the model will be unable to determine whether some content should be presented. Equally, a similar issue exists when discussing the representations of the tasks. Unless those tasks have been shown to a number of students, it is difficult to infer their relative difficulty.

In the study presented in Chapter 3, the vocabulary data used is split into CEFR levels. This provides an initial structure for the instructional content (the image-word pairs) and reduces the search space of possible optimal tasks for the student. However, not all tasks in language can easily or efficiently be divided into such a structure. The benefit of CEFR structure for vocabulary learning is that we know a priori that two words are considered to be at the same difficulty level. Search becomes a matter of finding the appropriate CEFR level and then the optimal word within that level. Assuming an equal number of words in every CEFR level, in the worst case scenario, we have reduced the complexity from $O(V)$ to $O(V/6) + 5$ where V is the number of words in the vocabulary. But can we

¹Note the terms ‘student’ and ‘user’ are used synonymously; as are ‘task’ and ‘item’.

improve on this? Is there a way to create representations of students that take advantage of the experience of other students? That is the main purpose of this chapter and study.

Previous approaches to constructing student representations include relying on manually engineering the features of the representations (Montero et al., 2018). These features are usually tuples of a knowledge component (e.g. differentiation and fractions, in the case where the domain is mathematics) and student outcome (i.e. whether or not the student demonstrated understanding for that knowledge component through completing the task). A task may contain multiple knowledge components. Whilst this approach is highly interpretable, in the domain of language learning, it is difficult to clearly divide the tasks into knowledge components. Therefore, it is of interest to explore methods of automatic construction of knowledge components.

Motivated by the success of representation learning within deep learning research, I present a methodology of automatically developing high quality representations of students and tasks in a language learning context. Similar to other popular techniques in deep learning, I allow the model to determine the necessary knowledge components (features) within the representation. Having reliable student and task representations in place will improve the accuracy of RL approaches to instructional sequencing. In addition, reliable representations can also yield interesting insights into how students learn language.

Representations are derived from a novel neural architecture (described in Section 4.4.2) and real student data collected through Write & Improve² (W&I), an assessment and feedback platform for learners of the English language (Yannakoudakis et al., 2018). Our representations take the form of *embeddings* – numeric vectors of a certain dimensionality, densely representing complex datasets.

Such representations enable us to draw upon established methods from *representation learning*³ including concatenating embeddings from different sources of information, learning representations of different targets (in our case, users and tasks) and passing the resultant vectors to multi-layered neural networks to train prediction models for unseen

²<https://writeandimprove.com>

³An area of research that focuses on developing representations of data for deep learning tasks.

data.

To develop our student representations I incorporate information about a student’s essay submissions to W&I, their essay score history, and the grammatical errors made in those essays – all together an approximation of the student’s knowledge state for language learning at any given point. Our task representations, on the other hand, incorporate the aforementioned information amalgamated for all the students who have attempted a particular task. The reason for this design choice is motivated by the view that the difficulty of a task is defined by the way students interact with the task. I further constrain the task embeddings by training them to predict their respective difficulty level (beginner, intermediate and advanced).

I evaluate the quality of our student and task representations extrinsically: 1) I use a combination of student and task representations to predict a student’s overall score on a given task; 2) I use the student and task representations to predict the grammatical errors a student will make on a given task. The first task is a conventional one in educational data mining (Koedinger et al., 2015); the second tests the generalisability of the student representations by evaluating their aptitude for *transfer learning* – the application of machine models trained on one problem onto a different but related problem (Pan and Yang, 2010).

Our best-performing neural network model incorporates grammatical error distributions detected by ERRANT (Bryant et al., 2017) as a feature and achieves mean squared error (MSE) of 1.195 on score prediction, an absolute value of 1.093 on a scoring scale of 0-13. On the second task of predicting grammatical errors on an unseen task, I achieve a cosine proximity score of -0.426 (-1 being perfect alignment). These results support the use of grammatical error distributions as a feature to determine student ability.

Our main contributions are as follows:

- The introduction of a novel neural framework that can be used to automatically learn student and task representations for language learning without explicitly modelling knowledge components. As far as I know, this is the first model that uses neural

collaborative filtering to model student knowledge states.

- The incorporation and evaluation of automatically detected grammatical error representations as a key feature in our neural network classification model to learn user and task representations. When tested on an unseen task, our set-up yields reliable prediction of both user-task score as well as grammar errors made by students on tasks.

4.2 Related work

Our general objective is modelling the acquisition of procedural knowledge (Corbett and Anderson, 1994), and we can usefully envisage this as the successful learning of ‘knowledge components’ (KCs) for any given educational domain (Koedinger et al., 2012). Models which take knowledge components into account have been shown to trace learning more successfully than otherwise (Cheng et al., 2016, Guo et al., 2017).

Personalisation in educational technology is of wide interest, since learners are known to progress at different rates and in different styles (Bloom, 1970, Saettler, 2004, Sampson and Karagiannidis, 2002, Ba-Omar et al., 2007, Brinton et al., 2015). Without an ontology or other knowledge base to guide personalisation (Tarus et al., 2018), we can only represent users through their interaction with learning items (tasks). Whereas well-known recommendation systems may have access to user ratings, reviews, click-throughs and sales figures, our measure of success is user performance – the score assigned to a given essay submission on the proposed item – and representation quality (predicting score and grammar errors on a task using the same representations).

Tracking users as they acquire knowledge in a learning system is a type of knowledge tracing, and previous approaches to knowledge tracing have ranged from item-response theory (Wilson et al., 2016), to Bayesian knowledge tracing (Corbett and Anderson, 1994), to deep learning (Montero et al., 2018), factorisation (Vie and Kashima, 2019) and dynamic time warping (Shen and Chi, 2018). I adopt a deep learning approach but, whereas for Montero et al. (2018) there were defined KCs in the mathematics domain

(e.g. fractions, differentiation) which could each be assigned binary values representing whether the student got that KC right for a question, in language learning it is not so clear how KCs should be defined and delimited. Therefore, I rely on learning representations through interaction and back-propagation with respect to the score assigned to each essay as well as the grammatical error distributions of that essay.

Our work also has similarities to RECOMMENDER SYSTEMS that use *collaborative filtering* – a method of selecting items for an individual based on their history in relation to others’ histories (hence they are ‘collaborative’) (Breese et al., 1998, Dron et al., 2000, Sarwar et al., 2001). Recommender systems represent the general task of promoting items from a pool to an individual in ways which will be familiar to users of Amazon, Netflix, Twitter and so on (Munro et al., 1999, Lawrence and Urtasun, 2009, Burke, 2002, Deshpande and Karypis, 2004): For online retailers the item bank is a set of products; for streaming services it is a library of movies and programmes; and for social networks it is user-generated content. In our case the item bank is a curated pool of tasks for English language learners. Our long-term goal is to present a personalised curriculum to each learner, navigating them through the item bank in an optimal fashion.

Until recently, the standard approach to collaborative filtering was matrix factorisation (MF) – most commonly with K nearest neighbours or singular value decomposition (SVD) – on user vectors containing interactions between the user and a set of items (Koren, 2008, Lawrence and Urtasun, 2009, Hu et al., 2014, He et al., 2016). Since then, novel approaches have shown that neural networks can improve both feature representation and collaborative filtering for recommender systems, with for instance, He et al. (2017) reporting 5-point hit-ratio gains over state-of-the-art MF systems using a multilayer perceptron. Zhang et al. (2018) observe that deep neural networks are well suited to the recommender task, since they are (1) end-to-end differentiable and (2) provide suitable *inductive biases* catered to the input data type. This also makes neural networks very well suited for the task for constructing student representations. In the language learning domain, an MF component was combined with a feed-forward neural network with some success for the 2018 Duolingo Shared Task on Second Language Acquisition Modelling (Settles et al., 2018, Vie, 2018).

Lindsey et al. (2014a) were interested in the effects of personalisation on drop-out rates: they found that a personalised review system for course content yielded a 16.5% boost in retention rates over standard practice (massed study) and a 10% improvement over a one-size-fits-all strategy for spaced study.

The recommender system approach has some precedent in educational technology. Early systems tended to involve heuristics or social networks (Anderson et al., 2003, Recker et al., 2003) before collaborative filtering techniques were introduced (Tang and McCalla, 2005, Nadolski et al., 2009, Wang and Yang, 2012). However, at present I do not attempt to implement a recommender system; instead I firstly inspect the quality of the user representations. Others have found that MF collaborative filtering alone is error-prone (Toscher and Jahrer, 2010). Various modifications to the standard model have been proposed, including question representations through difficulty rankings (Segal et al., 2014), fuzzy cognitive modelling (Wu et al., 2015), and ensemble models (Pardos and Heffernan, 2010).

To improve our student and task representations I incorporate automatically detected grammatical errors made on a task by a given student as a feature in our neural network model. Grammatical error detection is a well-established research field, with most focus having been placed so far on learners of English. Error detection techniques range from feature-based classification to neural machine translation (Rozovskaya et al., 2017, Yannakoudakis et al., 2017), and widely-used annotated corpora include the First Certificate in English (FCE) corpus (Yannakoudakis et al., 2011), the National University of Singapore Corpus of Learner English (Dahlmeier and Ng, 2012), and the JHU FLuency-Extended GUG corpus (Napoles et al., 2017). These corpora all involve different error typologies and one advantage of using ERRANT is that it defined a new error typology independent of but compatible with existing annotated data.

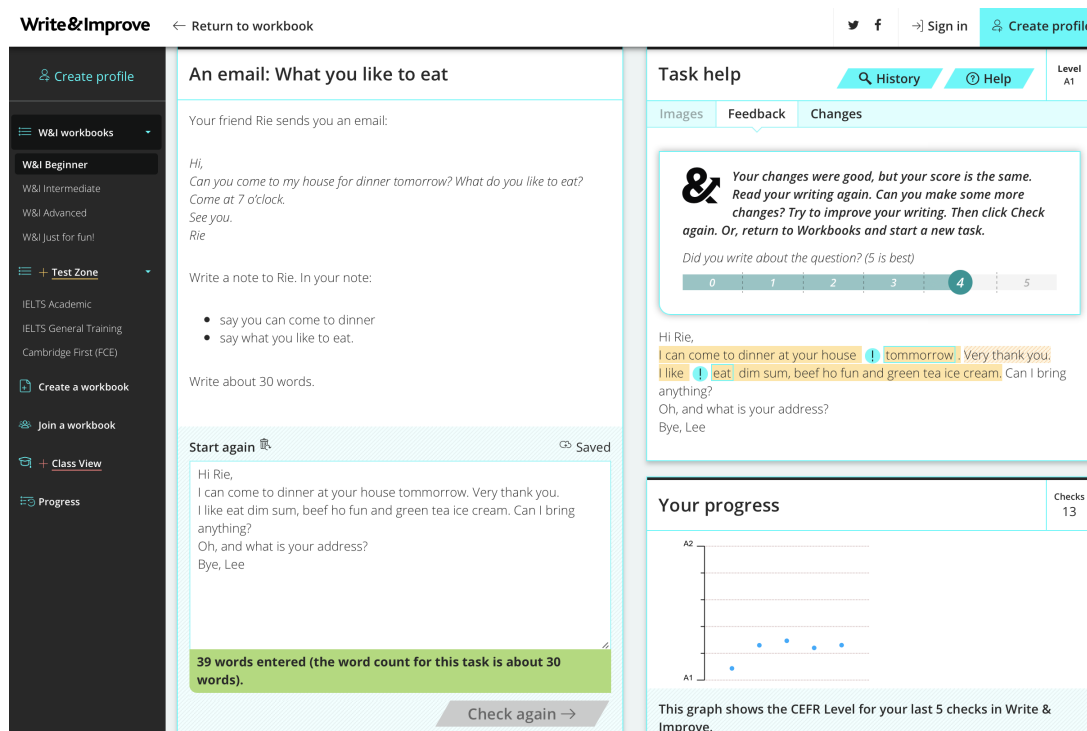


Figure 4.1: Write & Improve example screenshot

4.3 Write & Improve

On W&I, students are prompted to input a short text of at least 25 words in response to a given question. Once they have completed the task, the system automatically provides a grade on the CEFR scale along with feedback on grammatical errors detected in the text. The W&I automarker assigns each text an integer score between 0 and 13. Table 4.1 outlines how essay scores are mapped to the CEFR scale.

CEFR	Score
A1	1-2
A2	3-4
B1	5-6
B2	7-8
C1	9-10
C2	11-13

Table 4.1: Student scores mapped to CEFR levels

For instance, a student may submit a text such as that in (1) below, for which they

receive a score of 1.5, which equates to a grade of A1 (beginner) and indications that *tommorrow* and *I like eat* are ungrammatical. A screenshot of this example is provided in Figure 4.1.

1. Hi Rie,

I can come to dinner at your house tommorrow. Very thank you.

I like eat dim sum, beef ho fun and green tea ice cream. Can I bring anything?

Oh, and what is your address?

Bye, Lee

The student is encouraged to update and resubmit their text for further scoring and error feedback, and there is, in principle, no upper limit on the number of submissions they can make for a given task. It is their choice when to deem the task ‘complete’ and move on to a new question.

It is our long-term aim to develop an adaptive tutoring system (ATS) for language learners. There are 122 unique question items, or TASKS, in the W&I curriculum. Currently all users of W&I move through the curriculum in an unguided and independent fashion. An ATS would instead guide students from task to task in order to personalise their learning experience and improve their level of performance.

In order to provide this type of guidance, we need accurate representations of task difficulty and student ability as an essential prerequisite. W&I currently has tasks grouped by three broad difficulty levels: beginner, intermediate, advanced. However, our task representations need to be more fine-grained than this, so that we can guide students within and across the broader levels, and identify parts of the broad tripartite curriculum which have been separated *a priori* but are in fact of overlapping difficulty levels. Therefore I attempt to jointly train student and task representations based on past performance of real W&I users to capture the relative difficulty of tasks such that they can be reliably used to predict a particular student’s score on a given task.

4.4 Learning student and task representations

In this section, I will present the methods, approaches and datasets I developed or adapted for the purposes of this investigation. Our primary goal was to predict student scores on a given language learning task based on our representations of students and tasks in Write & Improve . Secondary to that, I check the quality of the student representations by predicting the grammar error distribution of a given student-task tuple. In what follows I describe the data, evaluation metrics and models used in this work.

4.4.1 Write & Improve data

Our training and test data come from the W&I language learning platform. W&I users submit responses that are at least 25 words in length for automated scoring and error feedback, and may opt to answer any number of prompts tagged with one of three difficulty levels – beginner, intermediate, advanced. I obtained application logs of user activity from 2017-2019 – a total of 3+ million essay submissions by 300,000+ account holders.

I filtered the data for users who had submitted at least 10 submissions. This resulted in a dataset of 1.3 million submissions by 100,140 users. I also had a record of the questions (‘prompts’) users responded to and the scores assigned to their texts by W&I’s auto-marker.

In addition, I obtained counts of grammatical errors in each submitted text using the ERRANT annotation toolkit Bryant et al. (2017). This gives us a distribution over 55 possible error types (See Appendix A), of which 47 were observed in the data we work with.

4.4.2 Model architecture

The architecture of our neural system can be seen in Figure 4.2. The neural network takes as an input a user id u and task id t which are taken as indices in the user embedding layer U and task embedding layer T respectively. $u \in N_u$ where N_u is the set of unique users in the W&I dataset. $t \in N_t$ where N_t is the set of unique tasks in the W&I dataset.

U is an $|N_u| \times d_u$ matrix where d_u is the size of the user representation \vec{u} . T is a

$|N_t| \times d_t$ where d_t is the size of the task representation \vec{t} . The description of the score prediction model can be seen in Equation 4.1:

$$\begin{aligned}
c &= (\vec{u}, \vec{t}) \\
h_1 &= \mathbf{D}(\sigma(c \cdot W^1)) \\
h_2 &= \mathbf{D}(\sigma(h_1 \cdot W^2)) \\
s &= h_2 \cdot W^s
\end{aligned} \tag{4.1}$$

– where c is the concatenated vector of features, h_1 and h_2 are the first and second hidden layers, \mathbf{D} is the dropout function (Srivastava et al., 2014) and σ is the ReLU activation (Nair and Hinton, 2010). W^1, W^2, W^s are the weight parameters of the model. Finally, s is the predicted score of user u on task t .

I optimise our system and learn a user embedding matrix U and task embedding matrix T by minimising the mean squared error (MSE) of our predicted score s and the target score \hat{s} :

$$L = \frac{1}{K} \sum_k (s - \hat{s})^2 \tag{4.2}$$

– where k is a given submission by the user for a particular task and K is the total number of submissions.

I introduce an auxiliary objective to predict the difficulty β of each task t , referenced as t_β . The ground-truth labels for task difficulty (beginner, intermediate, advanced) are obtained from the meta-data of each task in the dataset:

$$\begin{aligned}
h_3 &= \mathbf{D}(\sigma(\vec{t} \cdot W^3)) \\
t_\beta &= \textit{softmax}(h_3 \cdot W^\beta)
\end{aligned} \tag{4.3}$$

– where h_3 is the hidden layer between the task embedding matrix T and the output and W^3 and W^β are the weight parameters. I optimise the prediction of task difficulty t_β using a categorical cross-entropy loss function:

$$\mathcal{L} = -\frac{1}{N_t} \sum_i \sum_\beta \cdot 1_{t_\beta \in \beta} \log p(t_\beta \in \beta) \quad (4.4)$$

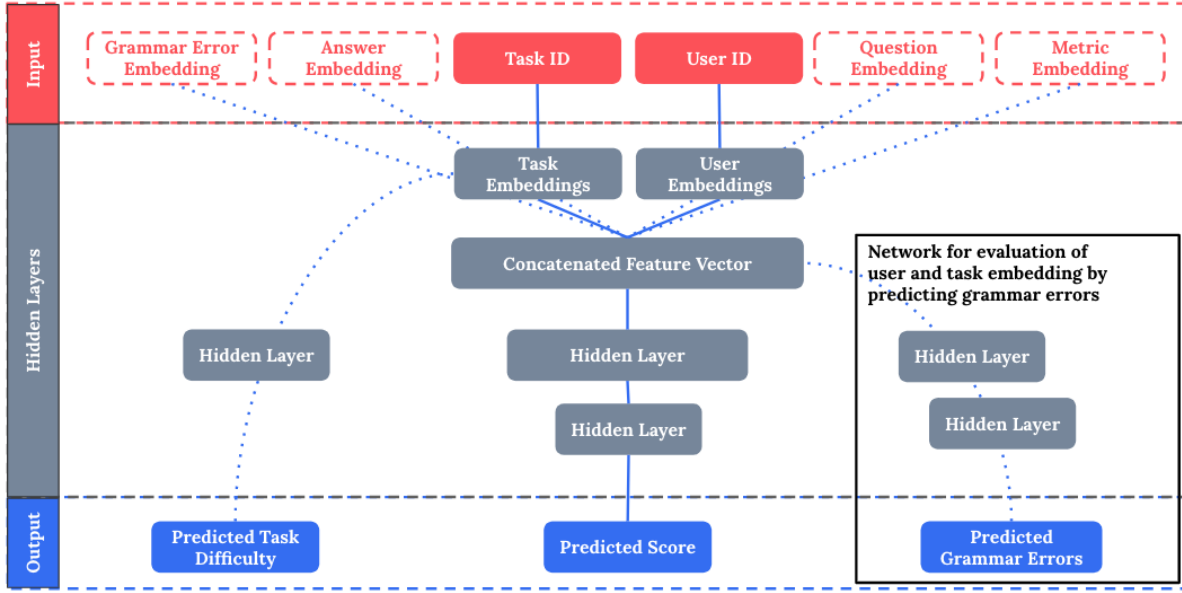


Figure 4.2: Task score prediction system architecture. Dotted lines and boxes are optional features and network connections.

4.4.3 Feature set

In addition to the score s , the W&I dataset contains prompts and answers in natural language as well as metrics on whether submission k is the highest scoring submission by user u . I incorporate these additional features into the architecture of the model in order to evaluate their impact on the quality of user and task embeddings.

4.4.3.1 Answer embedding

I obtain a vectorised form of each student response using 300-dimension word2vec embeddings⁴ pre-trained on the Google News corpus (Mikolov et al., 2013b). This means that we have information about the way words tend to be used by knowing which other words they are found to co-occur with, learned from a large dataset of news articles. In our case, the answer embedding for a student’s essay is an additive compositional model where the final embedding is a sum of every word in the essay. Whilst this model is not state-of-the-art for distributional semantics, Mitchell and Lapata (2010) show that the additive model can yield results comparable to significantly more sophisticated models. Additionally, whilst you might lose a lot of semantic information through such an approach, it remains practical for real-time usage.

4.4.3.2 Question embedding

Similar to the answer embeddings, I construct a vectorised form of each prompt represented in natural language, again summing word2vec representations of every word. I was motivated to incorporate question embeddings because I assumed that the lexical distribution of words in the prompt is directly correlated to the complexity of the question. I propose that linguistically complex questions are indicative of difficult tasks.

4.4.3.3 Metric embedding

The motivation behind using the metric embedding is to provide a signal to the model regarding the relative score of the submission in comparison to the user’s previous submissions. This signal may facilitate the model to down-weight submissions that are not task-best or user-best, as one could argue that task-best and user-best are a more accurate reflection of the student’s holistic capabilities.

The metric embedding is a 2-dimensional vector that stores benchmark information about submission k for user u in comparison to the user’s previous W&I submissions. The

⁴A word2vec embedding is a $1 \times x$ dimensional dense vector that represents a word semantically and according to its distributional properties. Words that are similar in meaning have vectors that are close together in vector space.

first dimension is a binary value for whether the score for the submission was the highest score on task t for user u . The second dimension is a binary value for whether the score for the submission was the highest score across all W&I tasks for user u . I have access to each user’s score history and infer metric embedding values by inspecting this history for each task and across all tasks.

4.4.3.4 Grammar error embedding

A student’s grammatical proficiency plays a vital role in determining how well they perform on a particular task. As we do not know of any system that identifies appropriate use of grammar, I focused on understanding what grammatical structures the student struggles with. This was done by running ERRANT (Bryant et al., 2017), an automated error detection and correction system, in order to identify grammatical errors in the student’s essay. The text below illustrates an example output from ERRANT.

S	Everythings	seem	quite meaningless to me .
A	0 1	R:SPELL	Everything REQUIRED -NONE- 0
A	1 2	R:VERB:SVA	seems REQUIRED -NONE- 0

The words highlighted in red are candidates for grammatical errors as detected by the system. The second and third lines are correction suggestions where the first two numerical digits (highlighted in blue) are the token spans for corrections (i.e. where in the sentence the corrections should apply); and the strings highlighted in green are the error types (e.g. R:SPELL, a spelling error; R:VERB:SVA, a subject-verb agreement error on the verb). ERRANT provides error detection and correction outputs on a sentence level. The last two columns of each line e.g. -NONE-|0 are not relevant for this task.

For each submission k , I constructed a 47-dimensional vector, one dimension for each of the error types observed in the W&I dataset. Each dimension stored the number of times that error type appeared in the student’s essay submission.

$$\langle g_k \rangle = \langle f_k^1, f_k^2, \dots, f_k^{47} \rangle \quad (4.5)$$

- where g_k is the grammar error embedding g for submission k , and f_k^n is the frequency of errors for error type n in submission k .

4.4.4 Mean score baseline

The baseline system for predicting an essay score s for user u on task t is to calculate the mean of observed scores by all users for that task. I refer to this baseline as MEAN_SCORE.

$$s_u^t = \frac{1}{N_k^t} \sum_k \hat{s}_k^t \quad (4.6)$$

- where s_u^t is the predicted score for user u on task t , N_k^t is the number of submissions for t , and \hat{s}_k^t is the observed score for submission k on t . Using MEAN_SCORE model, the predicted score on each task will be the same for all users.

Settles and Meeder (2016) showed that predicting the average is a strong baseline in modelling language learning – only 2 out of 4 models outperformed the average. Whilst the authors’ work focuses on predicting successful recall and understanding of words, I apply the same principal to the predicting student scores on unseen tasks.

4.4.5 Logistic regression baseline

A further baseline was constructed using a logistic regression model. This is motivated by the popularity of item response theory (IRT) in measuring student ability and task difficulty. IRT can be viewed as a special case of logistic regression.

The inputs for the logistic regression model were the student ability and task difficulty values. The output of the model was one of 13 possible discrete classes (0 to 13), representing the score a student achieved on a particular task.

I defined student ability α in this model as the student’s average score across all

attempted tasks n_i .

$$\alpha_i = \frac{\sum_{t=1}^T s_i^t}{n_i} \quad (4.7)$$

where α_i is the ability of student i , t is the task, s_i^t is the score student i obtained on task t and n is the number of tasks attempted by student i .

I defined task difficulty β_t as the average score achieved by all students on the given task t .

$$\beta_t = \frac{\sum_{i=1}^I s_i^t}{m_t} \quad (4.8)$$

where β_t is the difficulty of task t , s_i^t is the score student i obtained on task t and m is the total number of students that attempted task t

I then used a min-max scaler to transform the real values associated with student ability and task difficulty to a value between 0 and 1. Using a min-max scaler to bound the student ability and task difficulty enable the model to determine relative ability and difficulty more effectively.

I then trained a multiclass logistic regression model using cross-entropy loss in order in order to predict the score of an unseen student-task pair. In other words, given a student i and a task t , which has previously not been attempted, what would be the predicted score that student i would obtain on task t .

$$\frac{e^{xW+b}}{\sum_{i=1}^k e^{x_i W_i + b_i}} \quad (4.9)$$

where e is the Euler's mathematical constant, x is a vector of input features, in this model x contains the values of student ability α and task difficulty β . W is a weight matrix corresponding to the feature vector x , b is a bias vector, and k is the total number of classes, ranging from 0 to 13 in this is model.

4.4.6 Evaluation

I identify two approaches to evaluating our system and the quality of our learned user and task representations: 1) score prediction; and 2) grammar error prediction.

4.4.6.1 Evaluation of score predictions

To evaluate the performance of score prediction I use mean squared error (MSE) in common with other works in this field, using global computation where all data points are treated equally (Pelánek, 2018).

To form the test set, I remove the temporally last score observed by every student from our dataset. The last observed score, instead of a random observed score, was used due to the fact that as the student progresses through the learning material, both the student’s knowledge representation and the task representations evolve. Therefore, in order to ensure I am modelling the score that is based on the student’s most advanced knowledge state, I predict the last observed score for a student on a given task.

4.4.6.2 Evaluation of grammar embedding predictions

In order to further evaluate the quality of the learned user and task representations, I also introduce an additional evaluation task of predicting the distribution of grammar errors for a user u on a task t .

This was done by building a network that takes as an input the user \vec{u} and task \vec{t} from the pre-trained embedding U and T and predicts the grammar embedding \vec{g} . Our dataset for evaluating grammar error prediction was created by using ERRANT on the last submission k of every user u . This was to ensure that the system is predicting the distribution of errors for the users at their most recent knowledge state. The grammar

error embedding prediction model can be defined as follows:

$$\begin{aligned}
c &= (\vec{u}, \vec{t}) \\
h_1 &= \mathbf{D}(\sigma(c \cdot W^1)) \\
h_2 &= \mathbf{D}(\sigma(h_1 \cdot W^2)) \\
\vec{g} &= h_2 \cdot W^g
\end{aligned} \tag{4.10}$$

– where c is the concatenated vector of \vec{u} and \vec{t} , h_1 and h_2 are the first and second hidden layers, \mathbf{D} is the dropout function and σ is the ReLU activation function. W^1, W^2, W^g are the weight parameters of the model. \vec{g} is the predicted grammar error embedding for user u on task t .

I optimise the system by minimising the cosine proximity of the predicted grammar vector \vec{g} and the target grammar vector $\hat{\vec{g}}$, as in (4.11).

$$\mathcal{L} = - \frac{\sum_k \vec{g}_k \cdot \hat{\vec{g}}_k}{\sqrt{\sum_k (\vec{g}_k)^2} \cdot \sqrt{\sum_k (\hat{\vec{g}}_k)^2}} \tag{4.11}$$

– where k is a given submission by the user for a particular task. The more negative the cosine proximity the closer the prediction and target vectors. A value of -1 is a perfect match.

4.4.7 Implementation

I run the score prediction models for 30 epochs and use the Adam optimiser (Kingma and Ba, 2014) with a learning rate of 0.001. Both user embedding matrix U and task embedding matrix T were initialised with zero values. I initialise with zero values as we assume that we know nothing about the users and tasks at the beginning of the training session. In order to identify the right combination of features, I experiment with a variety of feature combinations and identify the ones that provide the greatest reduction in MSE

Features	N_h^{min}	N_h^{max}
<i>Score prediction model</i>		
user U	100,140 (of users) $\times 3$ (d_u)	$100,140 \times 32$
task T	122×3	122×32
answer	1×300	-
question	1×300	-
metric	1×2	-
error	1×47	-
h_1	1×8	-
h_2	1×4	-
h_3	1×3	-
<i>Grammar error prediction model</i>		
h_1	1×16	-
h_2	1×16	-

Table 4.2: Feature dimension sizes. N_h^{min} is the minimum size of the feature or only size where there is no value for N_h^{max} .

which will be discussed in Section 4.5. When evaluating our model at test time, I pass in null vectors for the metric, answer, and grammar error features as the student has, in theory, never attempted the task. Instead, I rely exclusively on the pre-trained user and task representations to make a reliable prediction of the user’s score s on task t .

For the grammar error prediction model I ran 50 epochs with an Adagrad optimiser (Duchi et al., 2011) and learning rate of 0.01. I used a dropout rate of 0.2 for both score prediction and grammar error prediction models.

Table 4.2 outlines the dimensions used for the various layers of the model. The user and task embedding were tested across a range of dimensions ranging from 3 to 32 dimensions. The justification behind using $n \times 3$ dimension embeddings was to align the size of the embedding with the number of task difficulty levels (beginner, intermediate and advanced). Furthermore, I created a bottleneck⁵ in our system in order to learn more meaningful student and task representations (Bengio et al., 2013). Therefore, I ensured that the upper-bound for the size of our user and task representations was less than 47 – that is, the number of dimensions in the smallest feature vector, the grammar error embedding⁶.

⁵A bottleneck is where the size of the representation layer is less than the size of the input.

⁶I excluded the metric embedding size as I assumed that an upper bound of 2 would not capture the

Model	MSE	Cosine
MEAN_SCORE (baseline)	1.913	-
logistic regression	1.296	
error+ques+ans+metric	2.254	-0.385
ques+metric	1.942	-0.402
ans+metric	1.951	-0.414
error+metric	1.350	-0.426
ques	2.028	-0.403
ans	2.014	-0.412
error	1.761	-0.410
metric	1.907	-0.393

Table 4.3: Score prediction (MSE) and grammar embedding prediction (cosine) results for the top 8 best performing feature combinations (error: grammar error embedding; ques: question embedding; ans: answer embedding; metric: metric embedding).

4.5 Results

Table 4.3 summarises the results of our system. I compare the effectiveness of various features in the prediction of a user’s score s on a task t which is evaluated by MSE. I include the top 8 MSE values on the score prediction system and their corresponding cosine value from the grammar error prediction model. Our baseline model MEAN_SCORE achieves an MSE of 1.913. The logistic regression model achieves an MSE of 1.296 which out performs the more sophisticated deep learning models.

I find that incorporating question and answer embeddings does not provide any performance improvement in terms of MSE beyond the baseline model. The metric embedding provides marginally better results than the baseline with an MSE of 1.907. The grammatical error embedding provides substantial improvements beyond both the baseline and the metric embedding with an error of 1.761. The best performing system incorporates both grammatical error embedding and metric embedding, reducing the MSE to 1.350.

The model that provides the lowest cosine proximity to the target grammatical error vector (i.e. best system) was error+metric, which is consistent with the lowest MSE for the score prediction system. I also observe that the system trained on just the answer feature

inherent complexity of language learning.

Model	N_h	MSE	Cosine
error+metric	3	1.350	-0.426
error+metric	5	1.297	-0.431
error+metric	16	1.245	-0.415
error+metric	32	1.195	-0.433

Table 4.4: Performance across various student and task representations sizes (N_h)

Pearson’s coefficient	p-value
0.7883	0.0201

Table 4.5: Correlation between score prediction MSE and grammar embedding prediction cosine.

resulted in a cosine proximity of -0.412 , an improvement over the system trained on just the grammar error embedding which achieves -0.410 . This outcome was unexpected: the system trained on the grammar error embedding resulted in a lower MSE than the system trained on the answer embedding, a representation which by definition contains the grammatical errors but not encoded in the same way. Intuitively the grammar error embedding is a better representation of student knowledge at a given point, which in turn gives us better predictions of task scores. Therefore, while the difference in MSE between answer embedding and grammatical error embedding is significant, their difference in cosine is not.

An important aspect of learning well-formed representations is identifying the correct number of dimensions Bengio et al. (2013). Table 4.4 summarises the various student and task representation sizes I used as part of the system. I set the upper bound at 32 in order to ensure a sufficient bottleneck. The results show that larger representation size improves both score prediction (MSE) and grammar error prediction (cosine).

In order to interpret the relevance of cosine proximity I conducted a Pearson’s correlation test between the MSE values from the score prediction system and the cosine proximity scores from the grammar error prediction system. Table 4.5 shows the correlation between the score predictions (MSE) and the grammar error prediction (cosine). The results show

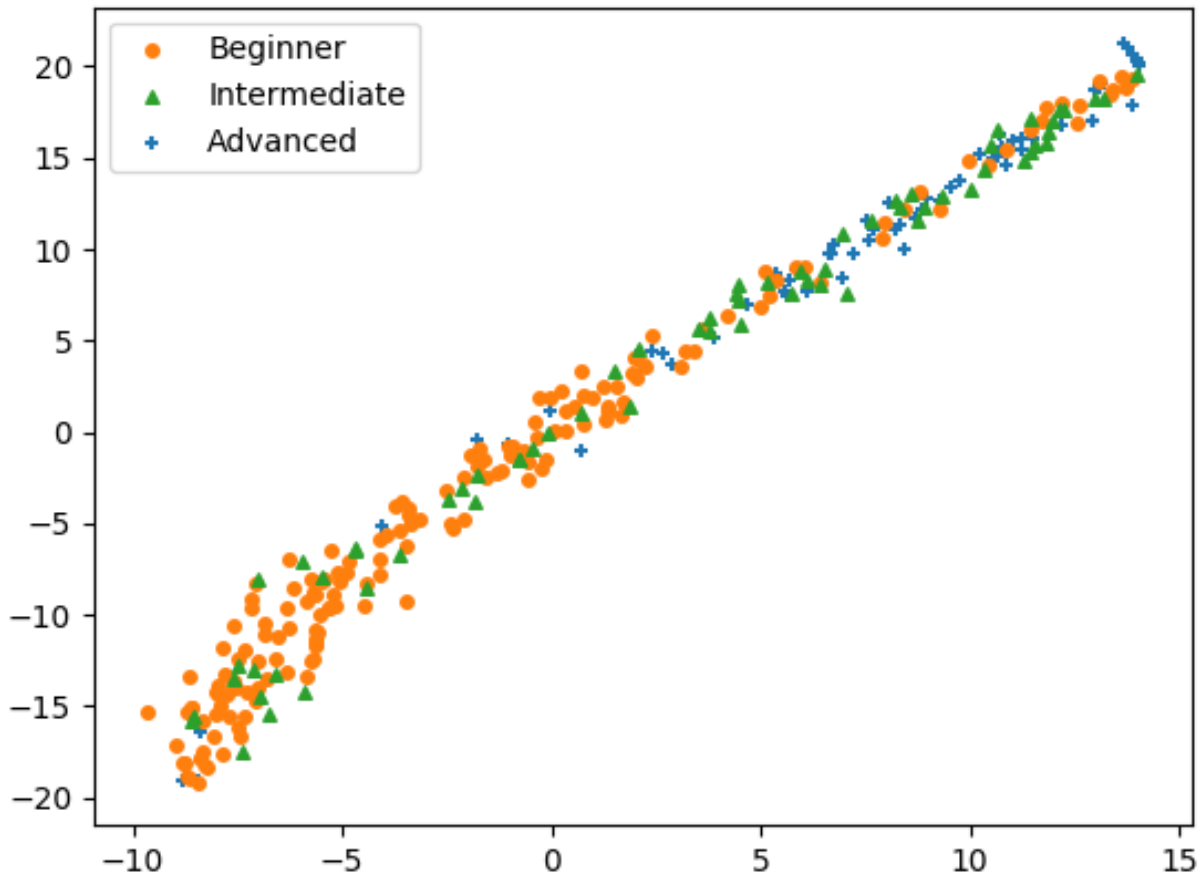


Figure 4.3: t-distributed stochastic neighbor embedding (t-SNE) of 300 randomly sampled student representations classified by different levels of proficiency

a 0.7883 Pearson’s correlation with a p -value of 0.0201 which is statistically significant at $\alpha < 0.05$.

Figure 4.3 shows a t-SNE (van der Maaten and Hinton, 2008) of 300 randomly sampled student representations learned by our best performing score prediction system. The students are classified by their proficiency which has been determined by observing the most frequent task level attempted in their five most recent submissions. Qualitatively, the results from the plot are promising as the advanced and intermediate users, whilst present throughout the plot, are more concentrated towards the top right (higher level of language proficiency). Beginner students, on the other hand, are more concentrated in the bottom left. This suggests that the embeddings constructed from our model provide context on the language abilities of the student.

4.6 Discussion

The results in Table 4.3 show that incorporating grammar error embeddings provides a reliable signal to learn well-formed student and task representations. Furthermore, Table 4.4 evaluates several dimensions for student and task representations by training the system using various configurations and evaluating both the MSE and cosine. Larger embedding size performed better than the smaller embedding sizes up to our experimental maximum of 32 dimensions. However, making the embedding size too large would result in what is known as ‘overcomplete’⁷ which in turn causes the model to simply memorise the correct response instead of learning discriminative features (Bengio et al., 2013).

In real terms, an MSE of 1.195 represents a root mean squared error of 1.093 on a scale of 0 to 13. This means that on average we stay within the bounds of a CEFR level when predicting student proficiency (since the 0-13 values are mapped to the 6-point CEFR scale), which seems sufficiently robust for real world application. The MSE might mask some more severe errors at the edges, and therefore any downstream use of our user and task representations for ATS would have to be implemented conservatively with reference to model confidence scores.

Grammar errors highlight the weaknesses of the student as opposed to their strengths. Therefore, instead of learning the upper-bound of a student’s ability, the model is learning the features for the lower-bound. The results of the model also suggest that there is a correlation between the types of errors students make on task t and the score they achieve on said task. This enables the model to learn latent features within the student and task representations which in turn can be used to reliably predict the student’s score on a future unseen task.

The importance and value of the signal provided by grammar errors in determining student ability and thus creating quality representations can be further highlighted by Figure 4.4. The bar-chart shows a comparison between beginner and intermediate students, where the values in x-axis are the various error types in ERRANT and the values for

⁷When $N_h > N_x$ (input).

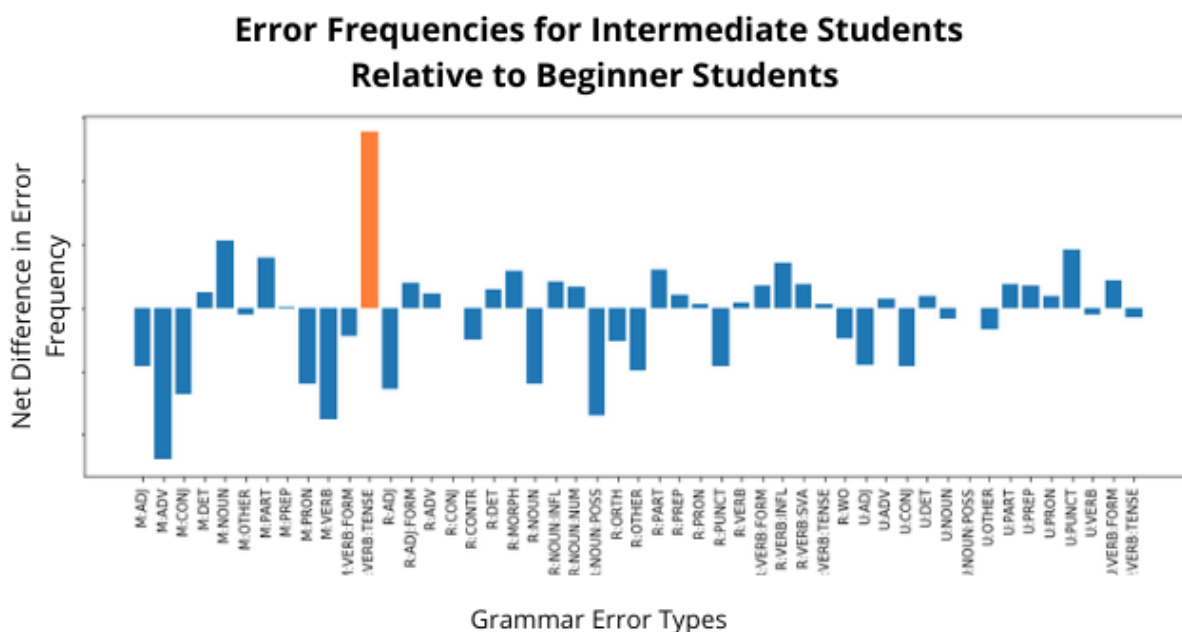


Figure 4.4: A bar-chart showing the frequency of error types for intermediate students, relative to beginner students. The orange bar indicates notable observations.

the y-axis are the normalised difference of the frequency for each error type between the two groups of students (positive bars indicate greater frequency of that error type for intermediate students). I can observe that certain errors such as **M: VERB: TENSE** (highlighted in orange) are more frequent with intermediate students. This is not surprising as beginner students tend not to experiment with verb tenses but rather focus on using verb tenses that they are comfortable with. Intermediate students are more likely to learn verb conjugation rules and over-regularise to introduce variation in sentence structure. However, over-regularisation usually results in increased number of verb tense errors (Rumelhart and McClelland, 1986, Bardovi-Harlig, 2000). This is then corrected once students reach an advanced level of proficiency where they can account for the irregular verb tenses. We can observe this correction in Figure 4.5 where advanced students make less verb tense errors than intermediate students.

I also show that a score prediction objective function with a task difficulty prediction auxiliary objective are effective in training well-formed student representations, as evidenced by Table 4.3, Table 4.4 and Figure 4.3. Whilst the plot in Figure 4.3 generally behaves as expected, we observe some students that are classified as *advanced* but reside towards the

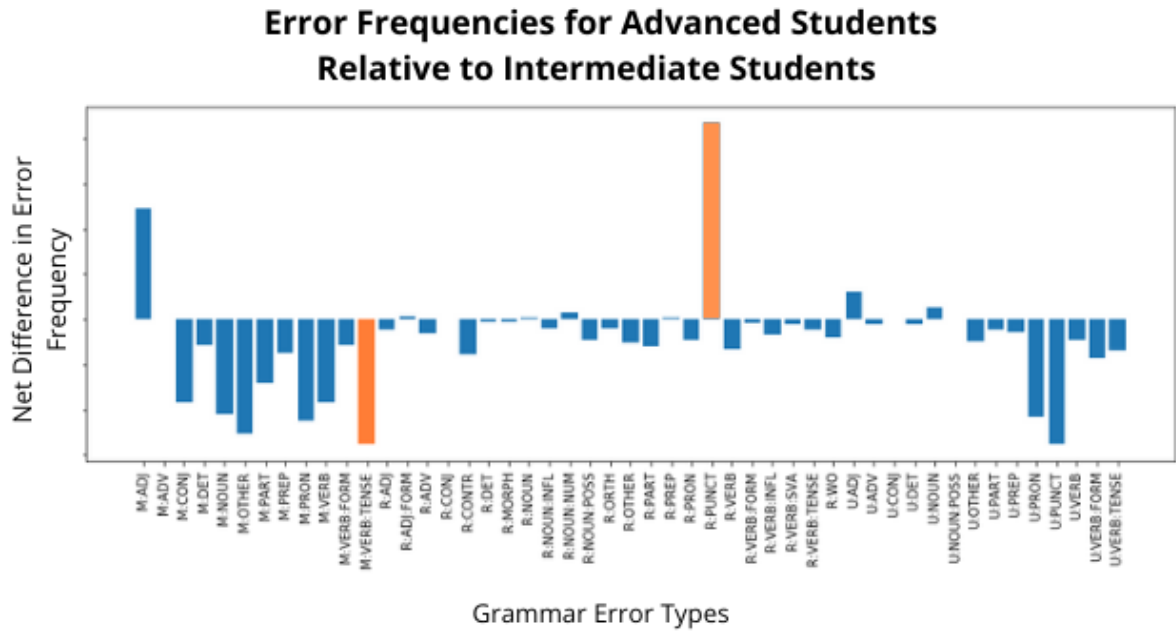


Figure 4.5: A bar-chart showing the frequency of error types for advanced students, relative to intermediate students. The orange bar indicates notable observations.

bottom-left. On inspection of the data, I believe this is due to students having *beginner* profiles but attempting *advanced* tasks –which they are at liberty to do.

Whilst grammar error features improved the performance of our system, answer embeddings seemed to reduce the accuracy, counter-intuitive to our understanding of the response as the fundamental indicator of student proficiency. I believe this reflects the fact that the answer text requires several levels of abstraction before it can be transformed into interpretable evidence of language proficiency. I therefore view grammar error embeddings as answer embeddings which have been passed through various levels of processing (in our case, by ERRANT (Bryant et al., 2017)). Without processing, answer embeddings provide only a noisy signal of student ability and negatively impact performance of predictive systems. The answers are relatively small samples of text, perhaps insufficiently so to properly trace language knowledge for the given student. Grammatical errors, on the other hand, appear to be sufficiently robust to short text lengths to provide representative signals of student knowledge. One additional reason for why the answer embedding does not provide a discriminative signal due to the fact that at test time a null vector is passed into the model. This results in a shift of distribution for the answer representations resulting

in downstream accuracy loss when predicting student score.

Question embeddings are faced with a similar limitation. I expressed the hypothesis that the wording of questions would directly indicate task difficulty. However, instead they proved to be the weakest standalone feature (Table 4.3). I interpret these findings together to mean either that it is how student’s perceive and respond to the question that determines the difficulty of the task, or that W&I scores are determined more by grammar errors than answer and question content. It is also possible that due to the presence of a task ID which the unique ID for every question, results in the question representation becoming redundant. Additionally, the content of the question itself does not yield any signal that is discriminative of student ability or task difficulty.

In isolation, the metric embedding also failed to provide a strong signal for student proficiency. However, combined with the grammar error embedding, I noticed significant improvements in performance.

The grammatical error distribution prediction system was introduced to further evaluate the quality of our student and task representations. The purpose of creating that system was to measure the generalisability of the student embeddings and demonstrate their ability to be leveraged for understanding the student’s strengths and weaknesses.

Although I do not know of an established gold standard for cosine proximity in our grammar error prediction task, I was able to interpret it in order to compare the performance between the different configurations of our user and task representation learning system. The positive correlation between the score prediction loss and the grammatical error prediction loss further supports our claim that our model architecture and the use of grammatical errors as features are reliable for training student and task representations of language learning. That is, the performance of the model is strong on two tasks, such that we view the representations of students and tasks as sufficiently accurate for further use in downstream educational applications.

4.7 Conclusion

I introduced a novel neural network model to automatically learn student and task representations for language learning by incorporating various features extracted from the W&I dataset and evaluating on score and grammar error prediction. I demonstrated through the results on the score prediction task that the use of grammar error embeddings and metric embeddings in our framework provide a reliable signal for user proficiency in language. These findings were further supported by the cosine proximity score achieved when evaluating the grammar error prediction task.

Learning user and task representations is a central component to enable a truly adaptive learning system. Future work in incorporating aspects such as memory decay and attention can play an important role in further improving the quality of user and task representations. Additionally, this framework may also enable downstream tasks such as curriculum learning in the language learning domain, item similarity (Pelánek et al., 2018), and task scheduling through spaced repetition learning (Mozer et al., 2009, Ling and Tan, 2018) .

Alongside the in-principle evaluation metrics I present here, I would like to be able to obtain real world evaluation of learning gains for trial groups presented with adaptively selected tasks, compared with control groups who continue to select tasks independently. I propose that the dense representations of users and tasks presented here could underpin an ATS which selects tasks at an appropriate difficulty level for each user with a known submission history on the platform.

Chapter 5

Human Forgetting Curves

In Chapter 3 and Chapter 4 I presented a method of personalised optimising instructional sequencing and developing student representations. However, neither of these contributions consider or implement the phenomenon of human forgetting within the design. More specifically, how in order to understand the optimal revision schedule, we must model how students forget content, specifically vocabulary words, over time. In this chapter, I consider how we can model forgetting specifically in relation to vocabulary acquisition.

This chapter was adapted from work published and presented in the 21st International Conference on Artificial Intelligence in Education (Zaidi et al., 2020).

5.1 Introduction

Optimal human learning techniques have been extensively studied by researchers in psychology (Dunlosky et al., 2013) and computer science (Settles and Meeder, 2016, Zaidi et al., 2017, Moore et al., Zaidi et al., 2019). The impact of learning techniques can be measured by how they affect the long-term retention of the learning materials. More specifically, better techniques will result in students retaining knowledge for longer periods of time.

Measuring retention requires a model of human forgetting, which describes the probability of recall over time. The first version of a “forgetting curve” was defined by Ebbinghaus

(Ebbinghaus, 1885a) but the idea has since been developed further by many researchers who have incorporated additional psychologically grounded variations to the model (Tabibian et al., 2019, Reddy et al., 2017, Mozer et al., 2019, Choffin et al., 2019, Rubin and Wenzel, 1996). While Ebbinghaus identified that our probability of recall declines exponentially with the passage of time, Melton (1970) and Dempster (1989), amongst others, investigated how to counter this phenomenon. The two key factors for maximising retention that emerged from these studies were: 1) frequency of review and 2) time lapsed between reviews. This became known as *spaced repetition*.

The rise of spaced repetition led to two new research questions: 1) how frequently to review content; 2) how far apart in time to space content review? This led to a range of heuristic systems including Pimsleur (1967) and Leitner (1972) systems. More recently, Metzler-Baddeley and Baddeley (2009) and Lindsey et al. (2014b) built upon these heuristic approaches by proposing systems that present content based on when the student is about to ‘forget’ i.e. when the probability to recall falls below a certain threshold. This threshold is calculated using a model of memory or forgetting curve e.g (Ebbinghaus, 1885b, Pashler et al., 2009, Settles et al., 2018).

The ideal forgetting curve and spaced repetition algorithm should adapt to learning materials as well as user meta-features (including current ability). Recently, with the rise of online learning platforms like Mnemosyne, Synap, and Duolingo we have seen an increased investment in the ability for technology to enable a more personalised approach for tasks like spaced repetition learning. However, as Tabibian et al. (2019) mentions, many of these approaches fall short of this promise. As a matter of fact most of these systems rely on rule-based heuristics with a few hard-coded parameters. Therefore, in this this study, I hope to investigate more sophisticated methods of modelling a forgetting curve. I examine this using the task of vocabulary learning and incorporate a range of linguistically motivated features, meta-features, and a variety of models in order to predict the probability a given learner will correctly recall a particular word. Our approach builds on the work of Settles et al. (2018).

The main contributions of the this chapter are as follows:

1. I present the state-of-the-art word recall prediction model using a simple neural network and psycholinguistic features. As far as we know, this is the first neural network based approach to modelling the forgetting curve.
2. I identify word-complexity as a valuable feature in predicting word recall. As far as we know, this is the first experiment that leverages e-learning platform data (i.e. Duolingo) to support the importance of word complexity is a feature in predicting recall rates.

5.2 Method

5.2.1 Duolingo Spaced Repetition Dataset

I use the Duolingo spaced repetition dataset (Settles, 2017) in order to train and evaluate our features and variety of models. The full dataset contains 13 million instances of student learning. When filtered for students learning English we are left with 4.28 million instances. Each data instance contains the following features, however not all of these features are used in the final model:

- `p_recall` –proportion of exercises from this lesson/practice where the word/lexeme was correctly recalled
- `timestamp` –UNIX timestamp of the current lesson/practice
- `delta` –time (in seconds) since the last lesson/practice that included this word/lexeme
- `user_id` –student user ID who did the lesson/practice (anonymised)
- `learning_language` –language being learned
- `ui_language` –user interface language (presumably native to the student)
- `lexeme_id` –system ID for the lexeme tag (i.e. word)

- `lexeme_string` –lexeme tag
- `history_seen` –total times user has seen the word/lexeme prior to this lesson/practice
- `history_correct` –total times user has been correct for the word/lexeme prior to this lesson/practice
- `session_seen` –times the user saw the word/lexeme during this lesson/practice
- `session_correct` –times the user got the word/lexeme correct during this lesson/practice

The models are an adaptation of the half-life regression model proposed by Settles and Meeder (2016) which was used for predicting the probability that a student will correctly recall a word. Typically the task presented to the student is a translation task where he/she will be presented with a Spanish word and the correct English word must be selected (or vice versa). Further details about the model proposed by the authors is discussed in Section 5.2.2.

I expand on the work of Settles and Meeder (2016) in three key ways: 1) I incorporate psycholinguistic features to capture more information about the words; 2) I use a simple neural network instead of an exponential function to compute the half-life \hat{h}_{Θ} value; 3) I incorporate a novel complexity rating for each word into the objective function for predicting probability of recall p as I believe that word complexity is directly linked to memory.

5.2.2 Half-Life Regression (HLR)

The half-life regression model proposed by Settles and Meeder (2016) is defined as follows:

$$p = 2^{-\Delta/h} \quad (5.1)$$

where p is the probability of recall, Δ is the time since last seen (days) and h is the *half-life* or strength of the learner’s memory. I denote the estimated half-life by \hat{h}_{Θ} , and it

is defined as:

$$\hat{h}_{\Theta} = 2^{\Theta \cdot \mathbf{x}} \quad (5.2)$$

where Θ is a vector of weights for the features \mathbf{x} .

The only feature used from the Duolingo spaced repetition dataset in the HLR model is the `lexeme_string` or lexeme tag. The lexeme tag takes the following form: `surface-form/lemma<pos>[<modifiers>...]`. For example, the lexeme tag for the word *camera* is `camera/camera< n >< sg >`, where `< n >` is noun and `< sg >` is singular. I assume that the authors' incorporation of lexeme tags into the model was motivated by the correlation between parts-of-speech and memory. This has been supported by Rodgers (1969) who found that when learning words in a foreign language, nouns are the easiest, adverbs are the most difficult, and verbs and adjectives are somewhere in the middle.

The HLR model is trained using the following loss function:

$$\ell(\mathbf{x}; \Theta) = (p - \hat{p}_{\Theta})^2 + (h - \hat{h}_{\Theta})^2 + \lambda \|\Theta\|_2^2 \quad (5.3)$$

In practice, Settles and Meeder (2016) found that optimising for both p and h in the loss function improved the model. The true value of h is defined as $h = \frac{-\Delta}{\log(p)}$. p and \hat{p}_{Θ} are the true probability and model estimated probability of recall, respectively. An L_2 -regularised squared loss was used which is denoted by $\lambda \|\Theta\|_2^2$ where λ is the constant used to control the regularisation term and prevent the model from overfitting.

5.2.3 HLR with Linguistic/Psychological Features (HLR+)

The model presented in Section 5.2.2 is a reimplementaion of previous work conducted by Settles and Meeder (2016). All the approaches and models presented henceforth were developed or adapted by myself for the purposes of this investigation.

I now expand on the HLR model by adding additional linguistic, psychological and meta-features to \mathbf{x} . I refer to this model as HLR+. The features include:

- **word complexity** –values are estimated by a pre-trained model Gooding and Kochmar (2019). The word complexity ranges from 0 to 1 where higher is more complex.
- **mean concreteness** –values are between 0 to 5 where higher is more concrete. Values were calculated based on human judgements found in Brysbaert et al. (2014).
- **percent known** –values range from 0 to 1 and are based on human judgements from Brysbaert et al. (2014)
- **SUBTLEX word frequencies** –the frequency of words based on a dataset that came out of the English Lexicon Project. Word frequencies range from 0 to 2.1 million (Van Heuven et al., 2014).
- **user ids** –the unique identifier for every student. Ranges from 0 to 43000. This data was extracted from the Duolingo spaced repetition dataset.

The motivation for including **word complexity** as a feature is based on the intuition that the more complex the word, the harder it is to remember. It is worth noting that word complexity is incorporated in different ways for different models (See Figure 5.2) Concreteness is included based on previous work showing that concrete words are easier to remember than abstract words because they activate perceptual memory codes in addition to verbal codes (Paivio, 2013). SUBTLEX represents the relative frequency of an English word based on a corpus of 201.3 million words: I hypothesise that more frequent words are more likely to be encountered and reinforced during the time since last seen Δ . Similarly, I expect that ‘percent known’ (the proportion of respondents familiar with each word based on survey data) will correlate with probability of recall. Lastly, I include user id to capture latent behavioural aspects about the learners.

5.2.4 Complexity-based Half-Life Regression (C-HLR+)

In addition to adding new features, I now describe a new model that modifies the p such that it directly incorporates *word complexity*. Word complexity can be defined in various

Feature	Range	Description
Word Complexity	0:1	Higher is more complex
Mean Word Concreteness	0:5	Higher is more concrete
Percent known	0:1	Higher is more known
SUBTLEX	0:2.1M	Frequency of word in corpus

Table 5.1: List of features and description in HLR+.

ways, but for the purposes of this work, it is defined as a relative measure of difficulty associated with learning the meaning of a word. The more difficult the word is to learn, the more “complex” the word is. In order to measure the difficulty of the word I relied on the state-of-the-art Complex Word Identification (CWI) system presented by Gooding and Kochmar (2019). The system outputs a value between 0 and 1. I hypothesise that this will correlate with probability of recall. As the complexity of the word rises, the forgetting curve will become steeper. Therefore, the new model is as follows:

$$p_i = 2^{-\Delta \cdot C_i / h} \quad (5.4)$$

where p_i is the probability of recall for word i and C is the mean complexity for word i . I define estimated half-life \hat{h}_Θ as $2^{\Theta \cdot \mathbf{x}}$ where \mathbf{x} is a vector composed of all of the features described in Section 5.2.3.

5.2.5 Neural Half-Life Regression (N-HLR+)

In my opinion, human forgetting is a complex phenomenon which most certainly cannot be reduced down to a single equation with a finite number of tangible features e.g. lexeme tag. Therefore, in order address this limitation I explore the use of neural networks. In recent years, neural networks have become increasingly popular in many complex tasks due to their ability to capture latent and abstract features. In theory, neural networks have the ability to capture features that might not be explicitly defined in the dataset through its multi-layer non-linear architecture. Therefore, in order to improve the estimation of half-life regression I describe a new model N-HLR+ model which replaces $\hat{h}_\Theta = 2^{\Theta \cdot \mathbf{x}}$ with

a neural network. The network can be described as follows:

$$\hat{h}_{\Theta} = ReLU(\mathbf{x} \cdot \mathbf{w}_1) \cdot \mathbf{w}_2 \quad (5.5)$$

where the network contains a single hidden layer. \mathbf{x} is a vector of input features, \mathbf{w}_1 is the weight matrix between the inputs and the hidden layer and \mathbf{w}_2 is the weight matrix between the hidden layer and the output. I use the same loss function as HLR which optimises for both p and h .

5.2.6 Evaluation and Implementation

I use mean absolute error (MAE) of probability of recall for a lexical item as the evaluation metric which, despite some known problems Pelánek (2015), is in line with previous work Settles and Meeder (2016). MAE is defined as: $\frac{1}{D} \sum_{i=1}^D |p - \hat{p}_{\Theta}|_i$, where D is the total data instances.

I divided the Duolingo English data into 90% training and 10% test. I trained all non-neural models (e.g. HLR, HLR+, C-HLR) using the following parameters which were tuned on the first 500k data points — learning rate: 0.001, alpha α : 0.01, λ : 0.1. For all neural models (e.g. N-HLR), I used — learning rate: 0.001, epochs: 200, hidden dimension: 4. I arrived at these parameters through a grid-search where I tested various permutations of learning rates and hidden dimensions. Due to the ‘overcomplete’ problem as described by Bengio et al. (2013), I ensured that $N_h > N_x$, where N_h is the hidden dimension and N_x is the input dimension. As the input dimension was 5 i.e. the number of features, I ensured that the hidden dimension was a maximum of 4.

5.3 Results and Discussion

We can see in Table 5.3 that HLR+, the model where I included psycholinguistic features, did not perform much better than HLR or HLR-lex. HLR-lex, a variation of HLR that excludes lexeme tags, yields better results than HLR suggesting that the lexeme tags only

Model	Description
HLR (Settles and Meeder, 2016)	Half-life regression model with lexeme tags
HLR-lex (Settles and Meeder, 2016)	Half-life regression model without lexeme tags
HLR+	Half-life regression model with psycholinguistic features
C-HLR+	Half-life regression model with psycholinguistic features and word complexity as a multiplicative factor
N-HLR+	Neural half-life regression model with psycholinguistic features
CN-HLR+	Neural half-life regression model with psycholinguistic features and word complexity as a multiplicative factor
NN	End-to-end neural network to predict probability or recall

Table 5.2: Summary of the various forgetting curve models.

add noise into the model.

By modifying the loss function to include complexity as a parameter as seen in the C-HLR+ model, I considerably improved the performance of the model. This was in line with the hypothesis that more complex words are forgotten faster and thus are an important feature in modelling the forgetting curve. It seems, that the original HLR model is unable to capture the relative importance of word complexity. This may be down to the simplicity of the model but also due to the noise contributed by the other, potentially less relevant, features such a SUBTLEX word frequency. Intuitively, word frequency might seem like a helpful feature in predicting recall i.e. words that appear more frequently are more likely to be remembered. However, finding a method of dealing with the large variance but also the continuous nature of the SUBTLEX feature is tricky and an area that requires additional investigation. One of the methods to reduce variance in the SUBTLEX feature was to *log* the word frequency value. Surprisingly, this did not yield better results in predicting probability of recall. One explanation is that there is

a strong correlation between word complexity and SUBTLEX frequency and as a result SUBTLEX frequency feature becomes redundant. It is possible that a more sophisticated technique of normalisation may yield better results with the word frequency feature.

The N-HLR+ model provided additional improvements to the C-HLR+ model. This supports the evidence that neural models are not only better at capturing non-linearities between the features and the expected output but also at capturing latent signals that are not explicitly referenced through features.

In the N-HLR+ model, I do not include a word complexity parameter in the objective function. Instead I only include word complexity in the feature vector when computing \hat{h}_{Θ} . In order to test the ability of neural networks to capture the latent feature I constructed an additional neural network (CN-HLR+) where I omit the word complexity from the feature vector and add it to the loss function (as I did in C-HLR+).

When compared to the N-HLR+ model to the modified CN-HLR+, I found that including complexity into the loss function in the CN-HLR+ provides no clear improvements in performance. This is because the neural model learns to place more importance on the *word complexity* and does not in anyway benefit from the word complexity parameter in the loss function as the C-HLR+ model did. I further confirm this by analysing the average weights in the hidden layer of the model. The model learns to give greater importance to word complexity, percent known, and concreteness respectively. It does not however, learn much from the user id and SUBTLEX. This is probably due to the fact that a single dimension for capturing user behaviour is not sufficient and as previously mentioned, SUBTLEX may require additional investigation in terms of how it can be incorporated into the feature vector.

The approach taken in this work was different in that it did not rely on end-to-end neural networks to address the task. Instead I relied on a hybrid approach, where part of the model was trained using a neural network and part of the model relied on domain understanding. I believe that this is why I was able to achieve state-of-the-art results on this task.

In order further investigate this claim, I trained an end-to-end neural network (denoted

Model	MAE↓	Model	MAE↓
Pimsleur (Pimsleur, 1967)	0.396	HLR+	0.129
Leitner (Leitner, 1972)	0.214	C-HLR+	0.109
Logistic Regression	0.196	N-HLR+	0.105
HLR (Settles and Meeder, 2016)	0.195	CN-HLR+	0.105
HLR-lex (Settles and Meeder, 2016)	0.130	NN	0.192

Table 5.3: Evaluation of forgetting curve models. Pimsleur and Leitner are previous methods of modelling the forgetting curve.

as the NN model in Table 5.3) that predicted \hat{p}_{Θ} directly from the feature vector. Using the same configuration as the N-HLR+ model in term of the number of neurons, I found that the NN model did not match the performance of N-HLR+ or CN-HLR+. In fact, I found that NN only marginally outperformed the original HLR model and performed worse than HLR-lex model. This illustrates the importance of incorporating domain-specific understanding in model selection. However, it is possible that with additional layers and more neurons an end-to-end neural model may match or outperform the N-HLR+ model.

5.4 Conclusion

I present a new model for adaptively modifying the forgetting curve for language learning using a modified HLR loss function and a neural network. The core contribution for this work is identifying the importance of word complexity as a feature in modelling recall probability. This idea can be extended beyond vocabulary learning by recognising that word complexity is simply a proxy for difficulty. Spaced repetition policies and associated memory models can benefit from some measure of difficulty. Furthermore, this work illustrates the importance of incorporating domain bias in defining the objective function. Settles and Meeder (2016) leverage their understanding of human forgetting and its underlying shape (i.e. exponential). This enabled them to design an objective function that was suited to the task. The results they presented yielded not only better performance when tested against other popular heuristic models, but also better student engagement and less drop off on their popular language learning app, Duolingo. Similarly,

I leveraged understanding of language, specifically that vocabulary should be taught from simple to complex, an idea that has been promoted throughout the history of personalised instructional sequencing (Saettler, 2004). Additionally, as we know from interference theory of memory and contextual cues, we are more likely to remember content which has been seen in a wide range of contexts (McGeoch, 1932). Therefore, it is not a surprising result that word complexity, which is correlated with word frequency (Kauchak, 2016), plays a significant role in defining the steepness of the forgetting curve. Furthermore, I also show a link between memory and word complexity that is supported by millions of data points, a contribution in its own right. While I know from Rodgers (1969) that word complexity and memory are related, to the best of my knowledge, this is the largest study (in terms of data points) that suggests word complexity is a good predictor of word recall and thus retention.

While the use of a simple neural network has shown to be a reliable method in predicting probability of recall for a vocabulary word, the results suggest that incorporating domain-bias such as a task specific objective function grounded in psychological theory is the key to that performance boost.

This work lays the foundation for work in neural approaches to understanding language learning over time. Future work in this area includes incorporating high-dimensional user embeddings as introduced in Chapter 4 to capture user specific signals that might influence the forgetting curve, and also different models such as Pareto and power functions which have been proposed in prior work Averell and Heathcote (2011). Building on contextual cues and memory, the role of a student's L1 (first language) is also an area that requires further investigation. Can we leverage information about the student's L1 to predict potential transfer learning and adapt the forgetting curve? Furthermore, understanding how to incorporate human forgetting to knowledge representations that ultimately influence RL-based personalised instructional sequencing policy introduced in Chapter 3 is also an important next step. Finally, the ultimate test for any machine learning model remains its ability to perform in a live environment. Therefore, integrating this system with a vocabulary learning environment to measure and monitor learning outcomes is also high

on the priority list for future work.

Chapter 6

Discussion and Conclusion

Personalised instructional sequencing has been at the centre of education since as far back as the Ancient Greeks. Although the term ‘personalised instructional sequencing’ was not coined in those days, the methodology of passing instruction from teacher to pupil was very personalised. Over the years, as education and teaching became more widespread and democratised, the growth in the number of teachers could no longer match the demand created by the growing number of students. Thus the one-to-one teaching experience was replaced with more homogenised forms of teaching and presenting instruction, for example the Lancastrian Model of Instruction. However, these methods went against the early and the more contemporary understanding of learning i.e. instruction should be adapted to the learner’s ability. This disconnect of theory and practice sparked the interest in scaling personalised instruction which was only accelerated with the advent of computers and later artificial intelligence. However, the challenge of technology enabled personalised instructional sequencing remains unsolved.

In this thesis, I tackle three specific components of technology-enabled personalised instructional sequencing: the mechanism of personalised instructional sequencing, the representations of the student which enables personalisation, and the human phenomenon of forgetting. This is not the entire landscape of personalised instructional sequencing but rather what I consider the foundation.

To emulate the behaviour of a teacher’s instructional sequencing decisions for a given

student, I use RL. RL has been used extensively for instructional sequencing but its impact remains unclear, especially because the comparative baseline used by many implementations of RL based personalised instructional sequencing experiments is randomised selection of content (Doroudi et al., 2019). Furthermore, Doroudi et al. (2019) concluded that the domains for which RL systems converged at a good instructional policy relied on psychological theory to drive their model selection. I observed this in my own work, where I used the structure of vocabulary words provided to me by CEFR. I also map different concepts with RL to personalised instructional sequencing with a novel approach to the concept of reward. Specifically, I propose that reward in a personalised instructional sequencing should relate to the long term benefit for the student and thus a reward of -1 should be given to the student for providing the correct answer. Considering the drill-and-practice systems developed by Atkinson and Raugh (1975), there are many similarities between my approach to reward presented in Chapter 3 and those systems. Namely, in drill-and-practice approach, there is no feedback provided for when the student enters the correct answer but rather only for when the student provides the wrong answer. The negative feedback can be seen as a method of increasing attention or importance by the student towards that item. Similarly, providing a student with a +1 reward for entering the incorrect answer is also increasing its relative importance. There is a lot of future work to be done in RL based personalised instructional sequencing systems but some of the areas that seem promising include combining psychological and cognitive theory with RL methods to improve policy selection and thus learning outcomes. Another possible extension is using RL models to uncover more about how human learning trajectory.

Without representation of student knowledge, implementing the “personalised” in personalised instructional sequencing is not possible. The question of how to model student knowledge for the purposes of inference (e.g. as a state in an RL system) remains an on-going discussion. One of the challenges of using RL is the explore-exploit problem. Yet, with enough iterations we have seen success on the use of RL based policies with video games (Mnih et al., 2013). However, in a video game setting, the state of the video game is information complete. All the information needed to make a decision is present

in the pixels of the game. On the other hand, when dealing with students, the state is their current knowledge state, which remains information incomplete and furthermore, the information we *can* obtain is through a noisy signal, usually an assessment event. This places additional focus on the importance of knowledge representation. Approaches to develop student knowledge representation vary from building a network of nodes where each node represents a KC in the domain to external state vectors, where each dimension of the vector is a KC. The challenge remains to identify the KCs in each domain. Let's consider a writing assessment. Typically a teacher will evaluate grammar, spelling, contextual relevance, and fluency when determining the level of the student. Spelling and grammar can in theory be broken down into vocabulary words and grammar rules/errors. On the other hand, breaking down fluency into its components is less straightforward. Additionally, representing misconceptions and transfer knowledge adds additional layers of complexity when developing knowledge representations. This creates the case for the use of deep learning as a method of learning representations instead of manually identifying what each dimension represents. This method of creating knowledge representation trades in interpretability for the possibility of richer representations that are able to learn high levels of abstractions; not an ideal trade-off.

In Chapter 4 I demonstrate how we can leverage approaches such as neural collaborative filtering to learn student knowledge representations. The motivation behind using such a method was motivated by the fact that since I am dealing with information incomplete knowledge states, leveraging information from similar students will allow me to boost the personalised instructional sequencing systems. I found that we can, with some degree of certainty predict how a student would perform on unseen tasks. This is valuable as it allows us predict the student's approximate ZPD and leverage the student representations as a state for my RL-based instructional sequencing system. The input of the neural collaborative filtering included the errors the student made in his/her answer. Upon analysing the errors, I identified that certain grammatical errors were far more common with intermediate students (e.g. verb tense errors). Intermediate students are more likely to over-regularise rules in grammar and as a result create more errors when applying rules

like verb conjugations.

We know that with the passage of time humans forget some information and remember other information. One of the challenges of personalised instructional sequencing is ensuring that information is reviewed frequently enough to prevent forgetting. Studies on human forgetting show that our recall of information decays at an exponential rate (Ebbinghaus, 1885b). In order to counter it there are strategies such as spaced repetition learning. But not all information is created equal and some items are forgotten quicker than others. In Chapter 5 I explore the role of word complexity in forgetting and whether we can leverage some measurement of word complexity to better predict when a student might forget a vocabulary word and thus optimise review strategy. Using word complexity I was able to achieve state of the art results in predicting probability of recall of a vocabulary word given the student. I also showed how using a simple neural network grounded in psychological theory resulted in better performance than just using a naive neural network. However, despite the positive result, there remain many additional areas of exploration with memory and spacing. Some of those include the question of how to integrate the work I presented in Chapter 5 with the RL-based personalised instructional sequencing model and student knowledge representations. Additionally, the study presented was conducted on real students, but on data which was static. Deploying the model in a live environment and conducting pre and post-test studies to evaluate the RL-based personalised instructional sequencing model’s impact on learning outcomes is the natural next step. Exploring other methods of representing student knowledge, e.g. as a graph, is also an important permutation in evaluating learning outcomes. Finally, like humans, machines also suffer from catastrophic forgetting when dealing with sequential tasks (McCloskey and Cohen, 1989). There remains a question of whether spaced repetition can help machine learning algorithms tackle catastrophic forgetting and improve performance.

In this thesis, along with an overview of the history of personalised instructional sequencing, I present a range of machine learning based systems that build a foundation for future work in this space. It is my hope that we can continue to push the limits of these systems and test them in live environments. Furthermore, as a way forward, looking

at how to integrate these systems with the existing workflow and role of teachers is critical. Finally, it is my hope that we continue to use theories of learning as a way to inform model selection and design but also use models to inform, validate and evolve our theories of learning.

Bibliography

- Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005. doi: 10.1109/TKDE.2005.99.
- Michelle Anderson, Marcel Ball, Harold Boley, Stephen Greene, Nancy Howse, Daniel Lemire, and Sean McGrath. RACOFI: a rule-applying collaborative filtering system. In *Proceedings of IEEE/WIC COLA '03*, 2003.
- Karl J Astrom. Optimal control of markov processes with incomplete state information. *Journal of mathematical analysis and applications*, 10(1):174–205, 1965.
- Richard C Atkinson. Ingredients for a theory of instruction. *American Psychologist*, 27(10):921, 1972a.
- Richard C Atkinson. Optimizing the learning of a second-language vocabulary. *Journal of experimental psychology*, 96(1):124, 1972b.
- Richard C Atkinson. Mnemotechnics in second-language learning. *American psychologist*, 30(8):821, 1975.
- Richard C Atkinson and James F Juola. Factors influencing speed and accuracy of word recognition. *Attention and performance IV*, pages 583–612, 1973.
- Richard C Atkinson and Michael R Raugh. An application of the mnemonic keyword method to the acquisition of a russian vocabulary. *Journal of experimental psychology: Human learning and memory*, 1(2):126, 1975.

- Lee Averell and Andrew Heathcote. The form of the forgetting curve and the fate of memories. *Journal of Mathematical Psychology*, 55:25–35, 2011.
- H Ba-Omar, I Petrounias, and F Anwar. A framework for using web usage mining to personalise e-learning. In *Proceedings of the 7th IEEE International Conference on Advanced Learning Technologies*, 2007. doi: 10.1109/ICALT.2007.13.
- Kathleen Bardovi-Harlig. Tense and aspect in second language acquisition: form, meaning, and use. *Language Learning: A Journal of Research in Language Studies*, 50:1, 2000.
- Carole R Beal and Ivon Arroyo. The animalwatch project: Creating an intelligent computer mathematics tutor. 2002.
- Joseph Beck, Beverly Park Woolf, and Carole R Beal. Advisor: A machine learning architecture for intelligent tutor construction. *AAAI/IAAI*, 2000(552-557):1–2, 2000.
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.
- Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.
- Marc G Berman, John Jonides, and Richard L Lewis. In search of decay in verbal short-term memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 35(2):317, 2009.
- Reinaldo AC Bianchi, Carlos HC Ribeiro, and Anna HR Costa. Heuristically accelerated q-learning: a new approach to speed up reinforcement learning. In *Brazilian Symposium on Artificial Intelligence*, pages 245–254. Springer, 2004.
- Benjamin S Bloom. Learning for mastery. instruction and curriculum. regional education laboratory for the carolinas and virginia, topical papers and reprints, number 1. *Evaluation comment*, 1(2):n2, 1968.

- Benjamin S Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational researcher*, 13(6):4–16, 1984.
- Benjamin S Bloom et al. Taxonomy of educational objectives. vol. 1: Cognitive domain. *New York: McKay*, 20:24, 1956.
- Lois Bloom. *Language development: Form and function in emerging grammars*. Cambridge, MA: MIT Press, 1970.
- Ronald J Brachman and Hector J Levesque. Knowledge representation and reasoning. *Morgan Kaufmann Publishers, Massachusetts, US*, 9:9, 2004.
- John Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, 1998.
- Christopher Brinton, Ruediger Rill, Sangtae Ha, Mung Chiang, Robert Smith, and William Ju. Individualization for education at scale: MIIC design and preliminary evaluation. *IEEE Transactions on Learning Technologies*, 8:136–148, 2015. doi: 10.1109/TLT.2014.2370635.
- John S Brown, Richard R Burton, and Frank Zdybel. A model driven question-answering system for a cai environment. 1972.
- John Seely Brown and Richard R Burton. Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive science*, 2(2):155–192, 1978.
- John Seely Brown, Richard R Burton, and Alan G Bell. Sophie: A step toward creating a reactive learning environment. *International Journal of Man-Machine Studies*, 7(5): 675–696, 1975.
- Jerome S Bruner. *The process of education:[a searching discussion of school education opening new paths to learning and teaching]*. Vintage Books, 1960.
- Jerome S Bruner. The act of discovery. *Harvard educational review*, 1961.

- Jerome Seymour Bruner et al. *Toward a theory of instruction*, volume 59. Harvard University Press, 1966.
- Peter Brusilovsky. Efficient techniques for adaptive hypermedia. In *intelligent Hypertext*, pages 12–30. Springer, 1994.
- Christopher Bryant, Mariano Felice, and Ted Briscoe. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017. URL <http://aclweb.org/anthology/P17-1074>.
- Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior research methods*, 46(3): 904–911, 2014.
- Robin Burke. Hybrid recommender systems: survey and experiments. *User Modeling and User-Adapted Interaction*, 12:331–370, 2002. doi: 10.1023/A:1021240730.
- R. J. Campbell, W. Robinson, J. Neelands, R. Hewston, and L. Mazzoli. Personalised learning: Ambiguities in theory and practice. *British Journal of Educational Studies*, 55 (2):135–154, 2007. ISSN 00071005, 14678527. URL <http://www.jstor.org/stable/4620550>.
- Jaime R Carbonell. Ai in cai: An artificial-intelligence approach to computer-assisted instruction. *IEEE transactions on man-machine systems*, 11(4):190–202, 1970.
- Brian Carr and Ira P Goldstein. Overlays: A theory of modelling for computer aided instruction. Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB, 1977.
- Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. Wide and deep

- learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS)*, 2016. doi: 10.1145/2988450.2988454.
- Benoît Choffin, Fabrice Popineau, Yolaine Bourda, and Jill-Jênn Vie. DAS3H: modeling student learning and forgetting for optimally scheduling distributed practice of skills. In *Proceedings of The 12th International Conference on Educational Data Mining (EDM)*, 2019.
- Marilyn Cochran-Smith, Sharon Feiman-Nemser, D John McIntyre, and Kelly E Demers. *Handbook of research on teacher education: Enduring questions in changing contexts*. Routledge, 2008.
- Johann Amos Comenius. *The Orbis Pictus of John Amos Comenius*. CW Bardeen, 1887.
- James Bryant Conant. *The education of American teachers*, volume 12387. New York: McGraw-Hill, 1963.
- Albert T. Corbett and John R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4:253–278, 1994. doi: 10.1007/BF01099821.
- Daniel Dahlmeier and Hwee Tou Ng. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2012. URL <http://www.aclweb.org/anthology/N12-1067>.
- Frank N Dempster. Spacing effects and their implications for theory and practice. *Educational Psychology Review*, 1(4):309–330, 1989.
- Mukund Deshpande and George Karypis. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems*, 22:143–177, 2004. doi: 10.1145/963770.963776.
- Paul B Diederich, John W French, and Sydell T Carlton. Factors in judgments of writing ability. *ETS Research Bulletin Series*, 1961(2):i–93, 1961.

- Shayan Doroudi, Vincent Aleven, and Emma Brunskill. Where's the reward? *International Journal of Artificial Intelligence in Education*, 29(4):568–620, 2019.
- Jon Dron, Richard Mitchell, Phil Siviter, and Chris Boyne. CoFIND – an experiment in n-dimensional collaborative filtering. *Journal of Network and Computer Applications*, 23:131–142, 2000. doi: 10.1006/jnca.2000.0105.
- John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul): 2121–2159, 2011.
- John Dunlosky, Katherine A Rawson, Elizabeth J Marsh, Mitchell J Nathan, and Daniel T Willingham. Improving students' learning with effective learning techniques: Promising directions from cognitive and educational psychology. *Psychological Science in the Public Interest*, 14(1):4–58, 2013.
- Herm Ebbinghaus. Ueber das gedächtnis. 1885a.
- Hermann Ebbinghaus. *Über das gedächtnis: untersuchungen zur experimentellen psychologie*. Duncker & Humblot, 1885b.
- CR Eliot, KA Williams, and BP Woolf. An intelligent learning environment for advanced cardiac life support. In *Proceedings of the AMIA annual fall symposium*, page 7. American Medical Informatics Association, 1996.
- J. Elliott. *Reconstructing Teacher Education: Teacher Development*. Falmer Press, 1993. ISBN 9780750701280. URL <https://books.google.co.uk/books?id=mi1g7ZHzeSgC>.
- Jeffrey L Elman. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99, 1993.
- Jean-Claude Falmagne, Mathieu Koppen, Michael Villano, Jean-Paul Doignon, and Leila Johannesen. Introduction to knowledge spaces: How to build, test, and search them. *Psychological Review*, 97(2):201, 1990.

- Dzikri Fudholi and Hanna Suominen. The importance of recommender and feedback features in a pronunciation learning aid. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 83–87. Association for Computational Linguistics, 2018. URL <http://aclweb.org/anthology/W18-3711>.
- Robert Mills Gagné, Robert Mills Gagne, et al. *Conditions of learning and theory of instruction*. Holt, Rinehart and Winston, 1985.
- Sian Gooding and Ekaterina Kochmar. Complex word identification as a sequence labelling task. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1148–1153, 2019.
- Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: a factorization-machine based neural network for ctr prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017. doi: 10.24963/ijcai.2017/239.
- Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In *SIGIR’16*, 2016. doi: 10.1145/2911451.2911489.
- Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *WWW’17*, 2017. doi: 10.1145/3038912.3052569.
- Karl Moritz Hermann, Felix Hill, Simon Green, Fumin Wang, Ryan Faulkner, Hubert Soyer, David Szepesvari, Wojciech Marian Czarnecki, Max Jaderberg, Denis Teplyashin, et al. Grounded language learning in a simulated 3d world. *arXiv preprint arXiv:1706.06551*, 2017.
- Ronald A Howard. Dynamic programming and markov processes. 1960a.
- Ronald A Howard. Machine-aided learning. *High speed computer system research: quarterly progress report*, 9:19–20, 1960b.

- Longke Hu, Aixin Sun, and Yong Liu. Your neighbors affect your ratings: On geographical neighborhood influence to rating prediction. In *SIGIR'14*, 2014. doi: 10.1145/2600428.2609593.
- Ana Iglesias, Paloma Martinez, and Fernando Fernández. An experience applying reinforcement learning in a web-based adaptive and intelligent educational system. 2003.
- Ana Iglesias, Paloma Martínez, Ricardo Aler, and Fernando Fernández. Learning teaching strategies in an adaptive and intelligent educational system through reinforcement learning. *Applied Intelligence*, 31(1):89–106, 2009.
- David Kauchak. Pomona at semeval-2016 task 11: Predicting word complexity based on corpus frequency. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1047–1051, 2016.
- Fred S Keller. Engineering personalized instruction in the classroom. *Revista Interamericana de Psicología/Interamerican Journal of Psychology*, 1(3), 1967.
- Mohammad Khajjah, Rowan Wing, Robert Lindsey, and Michael Mozer. Integrating latent-factor and knowledge-tracing models to predict individual differences in learning. In *Proceedings of Educational Data Mining (EDM)*, 2014.
- Sal Khan. Every child needs a personalised learning plan, Jan 2020. URL <https://www.wired.co.uk/article/personalised-learning-education>.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL <http://arxiv.org/abs/1412.6980>.
- Kenneth R Koedinger and John R Anderson. Illustrating principled design: The early evolution of a cognitive tutor for algebra symbolization. *Interactive Learning Environments*, 5(1):161–179, 1998.
- Kenneth R Koedinger, John R Anderson, William H Hadley, and Mary A Mark. Intelligent tutoring goes to school in the big city. 1997.

- Kenneth R Koedinger, Albert T Corbett, and Charles Perfetti. The knowledge-learning-instruction (kli) framework: Toward bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, 2010.
- Kenneth R Koedinger, Albert T Corbett, and Charles Perfetti. The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science*, 36(5):757–798, 2012.
- Kenneth R. Koedinger, Sidney D’Mello, Elizabeth A. McLaughlin, Zachary A. Pardos, and Carolyn P. Rosé. Data mining and education. *Wiley Interdisciplinary Reviews: Cognitive Science*, 6(4):333–353, 2015. doi: 10.1002/wcs.1350.
- KR Koedinger and ELF Sueker. Pat goes to college. In *Proceedings of the Second International Conference on the Learning Sciences*, 1996.
- Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. In *KDD’08*, 2008. doi: 10.1145/1401890.1401944.
- Stephen Krashen. We acquire vocabulary and spelling by reading: Additional evidence for the input hypothesis. *The modern language journal*, 73(4):440–464, 1989.
- Stephen D Krashen and Tracy D Terrell. The natural approach: Language acquisition in the classroom. 1983.
- Lev N Landa. The algo-heuristic theory of instruction. *Instructional-design theories and models: An overview of their current status*, 1:341–370, 1983.
- TK Landauer, RA Bjork, MM Gruneberg, PE Morris, and RN Sykes. Practical aspects of memory. *New York: Academic*, pages 625–632, 1978.
- Neil D Lawrence and Raquel Urtasun. Non-linear matrix factorization with gaussian processes. In *Proceedings of the 26th annual international conference on machine learning*, pages 601–608, 2009.

- Christian Lebiere and John R Anderson. A connectionist implementation of the act-r production system. 1993.
- Sebastian Leitner. *So lernt man lernen: angewandte Lernpsychologie—ein Weg zum Erfolg*. Herder., 1972.
- Robert V. Lindsey, Jeffery D. Shroyer, Harold Pashler, and Michael C. Mozer. Improving students’ long-term knowledge retention through personalized review. *Psychological Science*, 25(3):639–647, 2014a. doi: 10.1177/0956797613504302.
- Robert V Lindsey, Jeffery D Shroyer, Harold Pashler, and Michael C Mozer. Improving students’ long-term knowledge retention through personalized review. *Psychological science*, 25(3):639–647, 2014b.
- Ogden R Lindsley. Precision teaching’s unique legacy from bf skinner. *Journal of Behavioral Education*, 1(2):253–266, 1991.
- Lin Ling and Chee Wei Tan. Pilot study on optimal task scheduling in learning. In *Proceedings of Learning @ Scale*, 2018. doi: <https://doi.org/10.1145/3231644.3231677>.
- Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994.
- Kimberly N Martin and Ivon Arroyo. Agentx: Using reinforcement learning to improve the effectiveness of intelligent tutoring systems. In *International Conference on Intelligent Tutoring Systems*, pages 564–572. Springer, 2004.
- Laëtitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat. Hysteretic q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 64–69. IEEE, 2007.
- Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989.

- John A McGeoch. Forgetting and the law of disuse. *Psychological review*, 39(4):352, 1932.
- Arthur W Melton. The situation with respect to the spacing of repetitions and memory. *Journal of Verbal Learning and Verbal Behavior*, 9(5):596–606, 1970.
- Claudia Metzler-Baddeley and Roland J Baddeley. Does adaptive training work? *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition*, 23(2):254–266, 2009.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013b.
- Behrouz Minaei-Bidgoli, Deborah A. Kashy, Gerd Kortemeyer, and William F. Punch. Predicting student performance: an application of data mining methods with an educational web-based system. In *33rd Annual Frontiers in Education (FIE)*, 2003. doi: 10.1109/FIE.2003.1263284.
- Jeff Mitchell and Mirella Lapata. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429, 2010.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
- Shirly Montero, Akshit Arora, Sean Kelly, Brent Milne, and Michael Mozer. Does deep knowledge tracing model interactions among skills? In *Proceedings of the 11th International Conference on Educational Data Mining (EDM)*, 2018.
- Maria Montessori. The montessori method: Scientific pedagogy as applied to child education in ‘the children’s houses’ with additions and revisions by the author (ae george, trans.). *Cambridge, MA: Robert Bentley*, 1912.

- Russell Moore, Andrew Caines, Mark Elliott, Ahmed Zaidi, Andrew Rice, and Paula Buttery. Skills embeddings: a neural approach to multicomponent representations of students and tasks. In *Proceedings of The 12th International Conference on Educational Data Mining (EDM)*, volume 360, page 365. ERIC.
- Henry Clinton Morrison. *The practice of teaching in the secondary school*. University of Chicago Press, 1926.
- Michael C. Mozer, Harold Pashler, Nicholas Cepeda, Robert Lindsey, and Ed Vul. Predicting the optimal spacing of study: a multiscale context model of memory. In *Advances in Neural Information Processing Systems 22 (NIPS)*, 2009.
- Michael C Mozer, Melody Wiseheart, and Timothy P Novikoff. Artificial intelligence to support human instruction. *Proceedings of the National Academy of Sciences*, 116(10): 3953–3955, 2019.
- Alan Munro, Kristina Höök, and David Benyon. *Personal and social navigation of information space*. London: Springer-Verlag, 1999.
- Akira Murakami and Theodora Alexopoulou. L1 influence on the acquisition order of english grammatical morphemes: A learner corpus study. *Studies in Second Language Acquisition*, 38(3):365–401, 2016.
- Rob Nadolski, Bert van den Berg, Adriana Berlanga, Hendrik Drachsler, Hans Hummel, Rob Koper, and Peter Sloep. Simulating lightweight personalised recommender systems in learning networks: a case for pedagogy-oriented and rating based hybrid recommendation strategies. *Journal of Artificial Societies and Social Simulation*, 12, 2009. URL <http://jasss.soc.surrey.ac.uk/12/1/4.html>.
- Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 2010. URL <http://dl.acm.org/citation.cfm?id=3104322.3104425>.

- Amir Shareghi Najar, Antonija Mitrovic, and Bruce M. McLaren. Learning with intelligent tutors and worked examples: selecting learning activities adaptively leads to better learning outcomes than a fixed curriculum. *User Modeling and User-Adapted Interaction*, 26:459–491, 2016. doi: 10.1007/s11257-016-9181-y.
- Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. Jfleg: A fluency corpus and benchmark for grammatical error correction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2017. URL <http://www.aclweb.org/anthology/E17-2037>.
- NCEE. A nation at risk: The imperative for educational reform. *The Elementary School Journal*, 84(2):113–130, 1983.
- Allan Paivio. *Imagery and verbal processes*. Psychology Press, 2013.
- Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.
- Zachary Pardos and Neil Heffernan. Modeling individualization in a bayesian networks implementation of knowledge tracing. In *Proceedings of User Modeling, Adaptation, and Personalization*, 2010. doi: 10.1007/978-3-642-13470-8_24.
- Harold Pashler, Nicholas Cepeda, Robert V Lindsey, Ed Vul, and Michael C Mozer. Predicting the optimal spacing of study: A multiscale context model of memory. *Advances in neural information processing systems*, 22:1321–1329, 2009.
- Ivan Petrovich Pavlov. *The work of the digestive glands*. C. Griffin, 1910.
- Radek Pelánek. Metrics for evaluation of student models. *Journal of Educational Data Mining*, 7(2):1–19, 2015.
- Radek Pelánek. The details matter: methodological nuances in the evaluation of student models. *User Modeling and User-Adapted Interaction*, 28:207–235, 2018. doi: 10.1007/s11257-018-9204-y.

- Radek Pelánek, Tomáš Effenberger, Matěj Vaněk, Vojtěch Sassmann, and Dominik Gmiterko. Measuring item similarity in introductory programming. In *Proceedings of Learning @ Scale*, 2018. doi: <https://doi.org/10.1145/3231644.3231676>.
- Carol Peplinski. A generative cai program that teaches algebra. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 1970.
- Jean Piaget. Jan amos comenius. *Prospects*, 23(1-2):173–196, 1993.
- Paul Pimsleur. A memory schedule. *The Modern Language Journal*, 51(2):73–75, 1967.
- Anna N Rafferty, Emma Brunskill, Thomas L Griffiths, and Patrick Shafto. Faster teaching via pomdp planning. *Cognitive science*, 40(6):1290–1332, 2016.
- Mimi Recker, Andrew Walker, and Kimberly Lawless. What do you recommend? implementation and analyses of collaborative filtering of web resources for education. *Instructional Science*, 31:229–316, 2003. doi: 10.1023/A:1024686010.
- Siddharth Reddy, Sergey Levine, and Anca Dragan. Accelerating human learning with deep reinforcement learning. In *NeurIPS workshop: teaching machines, robots, and humans*, 2017.
- Georg Rehm, Peter Bourgonje, Stefanie Hegele, Florian Kintzel, Julián Moreno Schneider, Malte Ostendorff, Karolina Zaczynska, Armin Berger, Stefan Grill, Sören Räuchle, Jens Rauenbusch, Lisa Rutenburg, André Schmidt, Mikka Wild, Henry Hoffmann, Julian Fink, Sarah Schulz, Jurica Seva, Joachim Quantz, Joachim Böttger, Josefine Matthey, Rolf Fricke, Jan Thomsen, Adrian Paschke, Jamal Al Qundus, Thomas Hoppe, Naouel Karam, Frauke Weichhardt, Christian Fillies, Clemens Neudecker, Mike Gerber, Kai Labusch, Vahid Rezanezhad, Robin Schaefer, David Zellhöfer, Daniel Siewert, Patrick Bunk, Lydia Pintscher, Elena Aleynikova, and Franziska Heine. Quator: Innovative technologies for content and data curation, 2020.
- Frank E Ritter, Josef Nerb, Erno Lehtinen, and Timothy M O’Shea. *In order to learn: How the sequence of topics influences learning*. Oxford University Press, 2007.

- Theodore S Rodgers. On measuring vocabulary difficulty. an analysis of item variables in learning russian-english vocabulary pairs. *IRAL: International Review of Applied Linguistics in Language Teaching*, 7(4):327, 1969.
- Carolyn P Rosé, Elizabeth A McLaughlin, Ran Liu, and Kenneth R Koedinger. Explanatory learner models: Why machine learning (alone) is not the answer. *British Journal of Educational Technology*, 50(6):2943–2958, 2019.
- Yigal Rosen, Ilia Rushkin, Rob Rubin, Liberty Munson, Andrew Ang, Gregory Weber, Glenn Lopez, and Dustin Tingley. The effects of adaptive learning in a massive open online course on learners’ skill development. In *Proceedings of Learning @ Scale*, 2018. doi: <https://doi.org/10.1145/3231644.3231651>.
- Alla Rozovskaya, Dan Roth, and Mark Sammons. Adapting to learner errors with minimal supervision. *Computational Linguistics*, 43:723–760, 2017. doi: 10.1162/COLI_a.00299.
- David C Rubin and Amy E Wenzel. One hundred years of forgetting: A quantitative description of retention. *Psychological Review*, 103(4):734, 1996.
- David C Rubin et al. *Memory in oral traditions: The cognitive psychology of epic, ballads, and counting-out rhymes*. Oxford University Press on Demand, 1995.
- David E Rumelhart and James L McClelland. On learning the past tenses of English verbs. In David E Rumelhart, James L McClelland, and PDP Research Group, editors, *Parallel distributed processing: explorations in the microstructure of cognition, vol. 2*. Cambridge, MA: MIT Press, 1986.
- Stuart J Russell and Andrew Zimdars. Q-decomposition for reinforcement learning agents. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 656–663, 2003.
- Paul Saettler. *The evolution of American educational technology*. IAP, 2004.

- Demetrios Sampson and Charalampos Karagiannidis. Personalised learning: Educational, technological and standardisation perspective. *Interactive Educational Multimedia*, 4: 24–39, 2002.
- Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW’01*, 2001. doi: 10.1145/371920.372071.
- Avi Segal, Ziv Katir, Ya’akov (Kobi) Gal, Guy Shani, and Bracha Shapira. EduRank: A collaborative filtering approach to personalization in e-learning. In *Proceedings of the 7th International Conference on Educational Data Mining*, 2014.
- John A Self. Student models in computer-aided instruction. *International Journal of Man-machine studies*, 6(2):261–276, 1974.
- Burr Settles. Replication Data for: A Trainable Spaced Repetition Model for Language Learning, 2017. URL <https://doi.org/10.7910/DVN/N8XJME>.
- Burr Settles and Brendan Meeder. A trainable spaced repetition model for language learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1848–1858, 2016.
- Burr Settles, Chris Brust, Erin Gustafson, Masato Hagiwara, and Nitin Madnani. Second language acquisition modeling. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 2018. doi: 10.18653/v1/W18-0506.
- Shitian Shen and Min Chi. Clustering student sequential trajectories using dynamic time warping. In *Proceedings of Educational Data Mining*, 2018.
- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- Burrhus Frederic Skinner. *The behavior of organisms: An experimental analysis*. BF Skinner Foundation, 1938.
- Derek Sleeman and John Seely Brown. *Intelligent tutoring systems*. 1982.

- Richard D Smallwood. A decision structure for teaching machines. mit press research monograph, 14. 1962.
- Elliot M Soloway and William Lewis Johnson. Remembrance of blunders past: a retrospective on the development of proust. In *Proceedings of the sixth cognitive science society conference*, volume 57, page 60, 1984.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- Behzad Tabibian, Utkarsh Upadhyay, Abir De, Ali Zarezade, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. Enhancing human learning via spaced repetition optimization. *Proceedings of the National Academy of Sciences*, 116(10):3988–3993, 2019.
- Tiffany Ya Tang and Gordon McCalla. Smart recommendation for an evolving e-learning system. *International Journal on E-learning*, 4:105–129, 2005.
- John Tarus, Zhendong Niu, and Ghulam Mustafa. Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning. *Artificial Intelligence Review*, 50:21–48, 2018. doi: 10.1007/s10462-017-9539-5.
- Joel Tetreault and Diane Litman. Comparing the utility of state features in spoken dialogue using reinforcement learning. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 272–279, 2006.
- Edward L Thorndike, Elsie Oschrin Bregman, Margaret Vara Cobb, and Ella Woodyard. The measurement of intelligence. 1926.
- Edward Lee Thorndike. *The psychology of learning*, volume 2. Teachers College, Columbia University, 1913.

- Edward Lee Thorndike and Irving Lorge. The teacher's word book of 30,000 words. 1944.
- Edward Lee Thorndike, Elsie Oschrein Bregman, J Tilton, and Ella Woodyard. Adult learning. 1928.
- Andreas Toscher and Michael Jahrer. Collaborative filtering applied to educational data mining. In *KDD Cup*, 2010.
- Laurens J. P. van der Maaten and Geoffrey E. Hinton. Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9:579–2605, 2008.
- Walter JB Van Heuven, Pawel Mandera, Emmanuel Keuleers, and Marc Brysbaert. SUBTLEX-UK: A new and improved word frequency database for British English. *The Quarterly Journal of Experimental Psychology*, 67(6):1176–1190, 2014.
- Kurt VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.
- Jill-Jênn Vie. Deep factorization machines for knowledge tracing. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 2018. doi: 10.18653/v1/W18-0545.
- Jill-Jênn Vie and Hisashi Kashima. Knowledge tracing machines. In *Proceedings of AAAI*, 2019.
- Lev Semenovich Vygotsky. Socio-cultural theory. *Mind in society*, pages 52–58, 1978.
- Pei-Yu Wang and Hui-Chun Yang. Using collaborative filtering to support college students' use of online forum for english learning. *Computers & Education*, 59:628–637, 2012. doi: 10.1016/j.compedu.2012.02.007.
- Audrey Watters. The histories of personalized learning. <http://hackeducation.com/2017/06/09/personalization>, 2017.
- Jonathan David Wexler. A generative, remedial and query system for teaching by computer. 1970.

- Jacob Whitehill and Javier Movellan. Approximately optimal teaching of approximately optimal learners. *IEEE Transactions on Learning Technologies*, 11(2):152–164, 2017.
- Kevin H. Wilson, Yan Karklin, Bojian Han, and Chaitanya Ekanadham. Back to the basics: Bayesian extensions of IRT outperform neural networks for proficiency estimation. In *Proceedings of the 9th International Conference on Educational Data Mining (EDM)*, 2016.
- Charles P Winsor. The gompertz curve as a growth curve. *Proceedings of the National Academy of Sciences of the United States of America*, 18(1):1, 1932.
- Runze Wu, Qi Liu, Yuping Liu, Enhong Chen, Yu Su, Zhigang Chen, and Guoping Hu. Cognitive modelling for predicting examinee performance. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 2015. URL <https://www.aaai.org/ocs/index.php/IJCAI/IJCAI15/paper/view/11121>.
- Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011. URL <http://www.aclweb.org/anthology/P11-1019>.
- Helen Yannakoudakis, Marek Rei, Øistein E. Andersen, and Zheng Yuan. Neural sequence-labelling models for grammatical error correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017. URL <http://aclweb.org/anthology/D17-1297>.
- Helen Yannakoudakis, Øistein E. Andersen, Ardeshir Geranpayeh, Ted Briscoe, and Diane Nicholls. Developing an automated writing placement system for esl learners. *Applied Measurement in Education*, 31:251–267, 2018. doi: 10.1080/08957347.2018.1464447.
- Ahmed Zaidi, Andrew Caines, Russell Moore, Paula Buttery, and Andrew Rice. Adaptive forgetting curves for spaced repetition language learning. *arXiv preprint arXiv:2004.11327*, 2020.

Ahmed H. Zaidi, Russell Moore, and Ted Briscoe. Curriculum q-learning for visual vocabulary acquisition. In *Proceedings of Visually-Grounded Interaction and Language (ViGIL), NeurIPS*, 2017.

Ahmed H Zaidi, Andrew Caines, Christopher Davis, Russell Moore, Paula Buttery, and Andrew Rice. Accurate modelling of language learning tasks and students using representations of grammatical proficiency. In *EDM*, 2019.

Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys*, 1, 2018.

Appendix A

ERRANT Error Types

A list of error types outputted from ERRANT (Bryant et al., 2017)

		Type	Operation Tier		
			Missing	Unnecessary	Replacement
Token Tier	Part Of Speech	Adjective	M:ADJ	U:ADJ	R:ADJ
		Adverb	M:ADV	U:ADV	R:ADV
		Conjunction	M:CONJ	U:CONJ	R:CONJ
		Determiner	M:DET	U:DET	R:DET
		Noun	M:NOUN	U:NOUN	R:NOUN
		Particle	M:PART	U:PART	R:PART
		Preposition	M:PREP	U:PREP	R:PREP
		Pronoun	M:PRON	U:PRON	R:PRON
		Punctuation	M:PUNCT	U:PUNCT	R:PUNCT
		Verb	M:VERB	U:VERB	R:VERB
	Other	Contraction	M:CONTR	U:CONTR	R:CONTR
		Morphology	-	-	R:MORPH
		Orthography	-	-	R:ORTH
		Other	M:OTHER	U:OTHER	R:OTHER
		Spelling	-	-	R:SPELL
		Word Order	-	-	R:WO
Morphology Tier	Adjective Form	-	-	R:ADJ:FORM	
	Noun Inflection	-	-	R:NOUN:INFL	
	Noun Number	-	-	R:NOUN:NUM	
	Noun Possessive	M:NOUN:POSS	U:NOUN:POSS	R:NOUN:POSS	
	Verb Form	M:VERB:FORM	U:VERB:FORM	R:VERB:FORM	
	Verb Inflection	-	-	R:VERB:INFL	
	Verb Agreement	-	-	R:VERB:SVA	
	Verb Tense	M:VERB:TENSE	U:VERB:TENSE	R:VERB:TENSE	

Figure A.1: There are 55 total possible error types. This table shows all of them except UNK, which indicates an uncorrected error. A dash indicates an impossible combination.

